IBM Fusion



© Copyright IBM Corp. 2026.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Tables of Contents

Product overview 1
- Benefits of IBM Fusion 2
- IBM Storage Expert Care 3
- IBM Fusion trial version 3
- Fusion Essentials 3
- Security in IBM Fusion 3
- Verifying image signatures 4

What's new 5

Deploying IBM Fusion 7
- Prerequisites 8
  - Obtaining entitlement key 9
  - Creating image pull secret 9
  - Creating image pull secret for IBM Cloud based installation 10
  - IBM Fusion FIPS capabilities (Optional) 12
  - System requirements 12
  - SecurityContextConstraints 15
  - Enterprise registry for IBM Fusion installation 15
    - Mirroring prerequisites 16
    - End-to-end mirroring of IBM Fusion and its services images to the enterprise registry 16
    - Mirroring OpenShift Container Platform release images to enterprise registry for Hosted Control Plane 22
    - Mirroring Red Hat operator images to enterprise registry 23
    - Mirroring individual images 25
      - Mirroring IBM Fusion images 26
      - Mirroring IBM Storage Scale images 27
        - Mirroring IBM Storage Scale 6.0.1.0 images 29
      - Mirroring Fusion Data Foundation images 30
      - Mirroring Backup & Restore images 33
      - Mirroring IBM Data Cataloging images 35
      - Mirroring Content-Aware Storage (CAS) images 43
    - Troubleshooting and known issues in offline mirroring 44
- Installing IBM Fusion on On-premises VMware 46
- Installing IBM Fusion on On-premises Bare Metal 47
- Installing IBM Fusion on On-premises Linux on IBM Z and zCX 48
- Installing IBM Fusion on IBM Cloud 49
- Installing IBM Fusion on Amazon Web Services 50
- Installing IBM Fusion on Microsoft Azure 51
- Installing IBM Fusion on On-premises IBM Power Systems 52
- Installing IBM Fusion on Google Cloud 53
- Installing IBM Fusion from enterprise registry 53
  - Installing IBM Fusion On-premises 54
- Deploying IBM Fusion using CLI 55
- Validating IBM Fusion installation 56
- Uninstalling IBM Fusion 64
- Installation and upgrade issues 65

IBM Fusion Access for SAN 66
- Deploying IBM Fusion Access for SAN 67
  - Installing IBM Fusion Access for SAN operator 68
  - Creating a Kubernetes pull secret 69
  - Creating the FusionAccess custom resource (CR) 69
  - Creating a storage cluster 70
  - Creating a filesystem 70
  - Deploying in a disconnected (air-gapped) OpenShift environment 71
- IBM Fusion Access for SAN using Secure Boot 74

Storage 75
- Storage deployment combinations from the user interface 76
- Deploying your workloads with Persistent volumes 77
  - Static provisioning of Persistent Volume in filesystem 77
  - Dynamic provisioning of Persistent Volume in filesystem 78

Fusion services and capabilities 79
- Fusion services 80
  - IBM Fusion services and platform support matrix 81
  - Fusion Data Foundation 82
    - Release notes 82
      - New features 83
      - Enhancements 84
      - Known issues 85
    - Introduction to Fusion Data Foundation 89
    - Fusion Data Foundation architecture 89
      - An overview of Fusion Data Foundation architecture 89
      - Fusion Data Foundation Operators 90

Fusion Data Foundation operator 91
Components 91
Design diagram 91
Responsibilites 92
Resources 92
Limitation 92
High availability 92
Relevant config files 93
Relevant log files 93
Lifecycle 93
Container Storage operator 93
Components 93
Design diagram 93
Responsibilities 94
Resources 94
Limitation 95
High availability 95
Relevant config files 95
Relevant log files 95
Lifecycle 95
Rook-Ceph operator 95
Components 96
Design diagram 96
Responsibilities 97
Resources 98
Lifecycle 98
MCG operator 99
Components 99
Responsibilities and resources 99
High availability 100
Relevant log files 100
Lifecycle 100
Fusion Data Foundation installation overview 100
Installed Operators 101
OCSInitialization CR 101
Storage cluster creation 101
Internal mode storage cluster 102
Cluster creation 102
NooBaa System creation 103
External mode storage cluster 103
Cluster Creation 104
NooBaa System creation 103
Standalone Multicloud Object Gateway storage cluster 104
NooBaa System creation 103
Storage Cluster creation 105
Fusion Data Foundation upgrade overview 105
Upgrade Workflows 106
ClusterServiceVersion Reconciliation 106
Operator Reconciliation 106
Multus architecture for Fusion Data Foundation 106
Assumptions 107
Example scenario used 107
How to choose Multus architecture for a Fusion Data Foundation deployment 107
Multus public network 107
Multus public and cluster networks 108
Multus cluster network 109
Multus networking requirements 109
General requirements 109
Routing requirements 109
Address space sizing 110
Recommendations for Multus networking 110
General recommendations for Multus networking 110
Address ranges 110
Planning host interface connections 111
Example interfaces and connections 111
Recommended host interface connection 111
Multus network configuration 112
Public NetworkAttachmentDefinition 112
Cluster NetworkAttachmentDefinition 112
Node configuration for Multus public network 113
Preinstallation checklist for Multus configuration 113
Validating and debugging Multus configuration 114

Fusion Data Foundation installation after validating Multus configuration 114
Post-installation checks 114
Enabling Multus support on an existing Fusion Data Foundation cluster 115
Multus prerequisites validation tool 116
How to use validation tool 116
Troubleshooting validation tool 117
How to get help 117
Planning Fusion Data Foundation deployment 117
Storage cluster deployment approaches 118
Internal approach 119
Provider mode approach 119
External mode 119
Node types 119
Internal storage services 120
Provider storage services 120
External storage services 120
Security considerations 120
FIPS-140-2 121
Proxy environment 121
Data encryption options 121
Cluster-wide encryption 121
Storage class encryption 122
CipherTrust manager 122
Data encryption in-transit using messenger version 2 protocol of IBM Storage Ceph 122
Encryption in Transit 122
Subscription offerings 123
Cores versus vCPUs and hyperthreading 123
Cores versus vCPUs and simultaneous multithreading (SMT) for IBM Power 123
Splitting cores 123
Shared processor pools for IBM Power 124
Subscription requirements 124
Infrastructure requirements 124
Platform requirements 124
Amazon EC2 124
Bare Metal 125
VMware vSphere 125
Microsoft Azure 125
Google Cloud 125
Red Hat OpenStack Platform [Technology Preview] 125
IBM Power 125
IBM Z and IBM LinuxONE 125
Resource requirements 126
Resource requirements for IBM Z and IBM LinuxONE infrastructure 126
Minimum deployment resource requirements 127
Compact deployment resource requirements 127
Resource requirements for MCG only deployment 127
Resource requirements for using Network File system 128
Resource requirements for performance profiles 128
Pod placement rules 128
Storage device requirements 128
Dynamic storage devices 129
Local storage devices 129
Capacity planning 129
Network requirements 129
IPv6 support 129
Multi network plug-in (Multus) support 129
Disaster Recovery 130
Metro-DR 130
Disaster Recovery with stretch cluster 130
Regional-DR 130
Supported and unsupported features for IBM Power and IBM Z infrastructures 131
Installing Fusion Data Foundation 131
Uninstalling Fusion Data Foundation 133
Uninstalling Global Data Platform from Fusion Data Foundation external mode 134
Configuring Fusion Data Foundation 136
Configuring Fusion Data Foundation in local mode 138
Verifying Fusion Data Foundation local mode configuration 140
Verifying the state of the Pods 141
Verifying the Fusion Data Foundation cluster is healthy 141
Verifying the Multicloud Object Gateway is healthy 142
Verifying that the specific storage classes exist 142
Verifying the Multus networking 142

Configuring Fusion Data Foundation in dynamic mode 143
Verifying Fusion Data Foundation dynamic mode configuration 145
Verifying the state of the Pods 146
Verifying the Fusion Data Foundation cluster is healthy 147
Verifying the Multicloud Object Gateway is healthy 147
Verifying that the specific storage classes exist 147
Configuring Fusion Data Foundation in external mode 147
Configuring with IBM Storage Ceph 148
Verifying Fusion Data Foundation configuration with IBM Storage Ceph 150
Verifying the state of the Pods 151
Verifying the Fusion Data Foundation cluster is healthy 151
Verifying the Multicloud Object Gateway is healthy 151
Verifying that the specific storage classes exist 152
Applying in-transit encryption on IBM Storage Ceph cluster 152
Configuring with IBM FlashSystem 153
Verifying Fusion Data Foundation configuration with IBM FlashSystem 154
Verifying the state of the Pods 155
Verifying the Fusion Data Foundation cluster is healthy 156
Verifying the Multicloud Object Gateway is healthy 156
Configuring with IBM Storage Scale 156
Adding file systems to an IBM Storage Scale (CNSA) cluster 158
Removing a filesystem from an IBM Storage Scale (CNSA) cluster 159
Removing a remote IBM Storage Scale (CNSA) cluster 159
Configuring with IBM Fusion Access for Storage Area Network 159
Adding LUN groups to an IBM Fusion Access for SAN cluster 160
Configuring Fusion Data Foundation in provider mode 161
Verifying Fusion Data Foundation provider mode configuration 164
Configuring Fusion Data Foundation in MCG-only mode 165
Scaling out storage capacity 166
Adding a node to an installer-provisioned infrastructure 166
Adding a node to a user-provisioned infrastructure 167
Scaling up storage capacity 168
Enabling automatic capacity scaling 169
Preparing to connect to an external KMS server in Fusion Data Foundation 169
Enabling encryption with the token authentication using HashiCorp Vault(manual part) 170
Enabling encryption with the Kubernetes authentication using HashiCorp Vault (manual part) 171
Enabling key rotation for Kubernetes authentication 172
Disabling key rotation for Kubernetes authentication 173
Enabling encryption using Thales CipherTrust Manager (manual part) 174
Deploying Data Foundation in external mode 174
Deploying multiple Fusion Data Foundation storage clusters
Overview of multiple storage cluster deployments 177
Preparing to deploy multiple Fusion Data Foundation storage clusters 178
Deploying Data Foundation external storage cluster 178
Verifying external Data Foundation storage cluster deployment 180
Verifying the state of the pods 181
Verifying the Fusion Data Foundation cluster is healthy 181
Verifying the Multicloud Object Gateway is healthy 181
Verifying that the specific storage classes exist 182
Verifying that Ceph cluster is connected 182
Verifying that storage cluster is ready 182
Migrating application workloads 182
Managing and allocating resources 182
Storage classes 183
Creating storage classes and pools 183
Storage class with single replica 184
Persistent volume encryption 185
Access configuration for KMS 185
Configuring access to KMS by using vaulttokens 185
Configuring access to KMS by using Thales CipherTrust Manager 186
Configuring access by using vaulttenantsa 186
Creating a storage class for persistent volume encryption 188
Overriding Vault connection details using tenant ConfigMap 190
Enabling and disabling key rotation when using KMS 190
Enforcing precedence for key rotation 190
Enabling key rotation 190
Disabling key rotation 191
Enabling and disabling encryption in-transit post deployment 192
Enabling encryption in-transit after deployment in internal mode 192
Disabling encryption in-transit after deployment in internal mode 193
Enabling encryption in-transit after deployment in external mode 194
Applying encryption in-transit on Red Hat Ceph storage cluster 194

Remount existing volumes 195
Disabling encryption in-transit after deployment in external mode 195
Block pools 197
Block pools in internal mode 197
Creating a block pool 197
Updating an existing pool 197
Deleting a pool 198
Configure storage for OpenShift Container Platform services 198
Configuring Image Registry to use Fusion Data Foundation 198
Using Multicloud Object Gateway as OpenShift Image Registry backend storage 199
Configuring monitoring to use Fusion Data Foundation 201
Overprovision level policy control [Technology Preview] 202
Cluster logging for Fusion Data Foundation 203
Configuring persistent storage 204
Configuring cluster logging to use Fusion Data Foundation 204
Creating Multus networks 205
Creating network attachment definitions 206
Backing OpenShift Container Platform applications with Fusion Data Foundation 206
Adding file and object storage to an existing external Fusion Data Foundation cluster 207
How to use dedicated worker nodes for Fusion Data Foundation 209
Anatomy of an Infrastructure node 209
Managing persistent volume claims 209
Configuring application pods to use Fusion Data Foundation 210
Viewing Persistent Volume Claim request status 211
Reviewing Persistent Volume Claim request events 211
Expanding Persistent Volume Claims 211
Dynamic provisioning 212
Dynamic provisioning in Fusion Data Foundation 212
Available dynamic provisioning plug-ins 213
Reclaiming space on target volumes 213
Enabling reclaim space operation using Annotating Persistent Volume Claims 214
Disabling reclaim space for a specific PersistentVolumeClaim 214
Enabling reclaim space operation using ReclaimSpaceJob 215
Enabling reclaim space operation using ReclaimSpaceCronJob 215
Customizing timeouts required for Reclaim Space Operation 216
Finding and cleaning stale subvolumes 216
Volume snapshots 217
Creating volume snapshots 217
Restoring volume snapshots 218
Deleting volume snapshots 218
Volume cloning 219
Creating a clone 219
Managing Container Storage Interface (CSI) component placements 219
2-way replication with CephFS 221
Editing the existing default CephFS data pool to replica-2 222
Adding an additional CephFS data pool with replica-2 222
Creating exports using NFS 223
Enabling the NFS feature 223
Creating NFS exports 223
Consuming NFS exports in-cluster 224
Consume NFS exports externally from the cluster 225
Annotating encrypted RBD storage classes 226
Enabling faster client IO or recovery IO during OSD backfill 226
Setting Ceph OSD full thresholds 226
Setting Ceph OSD full thresholds by using the ODF CLI tool 226
Setting Ceph OSD full thresholds by updating the StorageCluster CR 227
Configuring Ceph target size ratios 227
Configuring Ceph monitor failover timeout 229
Backup and recovery of Multicloud Object Gateway metadata database (NooBaa DB) 230
Enabling backup of Multicloud Object Gateway metadata database using storagecluster CR 230
Enabling backup of Multicloud Object Gateway metadata database using MCG CLI 230
Recovering Multicloud Object Gateway metadata database from backup 230
Managing hybrid and multicloud resource 231
Accessing the Multicloud Object Gateway with your applications 231
Accessing the Multicloud Object Gateway from the terminal 232
Accessing the Multicloud Object Gateway from the MCG command-line interface 233
Support of Multicloud Object Gateway data bucket APIs 235
Adding storage resources for hybrid or Multicloud 235
Creating a new backing store 235
Overriding the default backing store 236
Adding storage resources for hybrid or Multicloud using the MCG command line interface 237
Creating an AWS-backed backingstore 237

Creating an AWS-STS-backed backingstore 238
Creating an AWS role using a script 238
Installing OpenShift Data Foundation operator in AWS STS OpenShift cluster 240
Creating a new AWS STS backingstore 240
Creating an IBM COS-backed backingstore 240
Creating an Azure-backed backingstore 242
Creating a GCP-backed backingstore 243
Creating a local Persistent Volume-backed backingstore 244
Creating an s3 compatible Multicloud Object Gateway backingstore 245
Creating a new bucket class 246
Editing a bucket class 246
Editing backing stores for bucket class 247
Creating and managing buckets using the object browser 248
Accessing the object browser 248
Signing in to the object browser as a developer user 248
Creating and managing buckets 249
Creating new buckets using the object browser 249
Listing bucket details using the object browser 249
Deleting or emptying bucket using object browser 250
Managing objects 250
Listing objects and object details using MCG object browser 250
Downloading or previewing objects using the object browser 250
Generating the presigned URL to share the object 251
Uploading objects using the object browser 251
Creating folder using the object browser 251
Deleting objects using the object browser 252
Enabling or suspending object versioning in a bucket using the object browser 252
Listing object versions using the object browser 252
Creating and managing lifecycle rules 253
Creating new lifecycle rules by using the object browser 253
Editing lifecycle rules by using the object browser 253
Creating CORS rule 254
Editing CORS rule 254
Applying bucket policy to grant public or restricted access to objects 254
Setting up public access limit to S3 resources 255
Creating and managing IAM user 255
Creating an IAM user using the object browser 256
Generating additional access key for an IAM user 256
Activating or deactivating access key 256
Deleting access key for an IAM user 257
Deleting an IAM user 257
Managing namespace buckets 257
Amazon S3 API endpoints for objects in namespace buckets 258
Adding a namespace bucket using the Multicloud Object Gateway CLI and YAML 258
Adding an AWS S3 namespace bucket using YAML 258
Adding an IBM COS namespace bucket using YAML 260
Adding an AWS S3 namespace bucket using the Multicloud Object Gateway CLI 262
Adding an IBM COS namespace bucket using the Multicloud Object Gateway CLI 263
Adding a namespace bucket using the OpenShift Container Platform user interface 264
Sharing legacy application data with cloud native application using S3 protocol 264
Creating a NamespaceStore to use a file system 265
Creating accounts with NamespaceStore file system configuration 265
Accessing legacy application data from the openshift-storage namespace 266
Changing the default SELinux label on the legacy application project to match the one in the openshift-storage project 271
Modifying the SELinux label only for the deployment config that has the pod which mounts the legacy application PVC 271
Configuring or modifying the PublicAccessBlock configuration for S3 bucket 272
Securing Multicloud Object Gateway 273
Changing the default account credentials to ensure better security in the Multicloud Object Gateway 273
Resetting the noobaa account password 274
Setting Multicloud Object Gateway account credentials using CLI command 274
Regenerating the S3 credentials for the accounts 275
Regenerating the S3 credentials for the OBC 276
Enabling secured mode deployment for Multicloud Object Gateway 277
Disabling routes that enable external access to Multicloud Object Gateway 278
Mirroring data for hybrid and Multicloud buckets 278
Creating bucket classes to mirror data using the MCG command-line-interface 278
Creating bucket classes to mirror data using a YAML 278
Bucket policies in the Multicloud Object Gateway 279
Introduction to bucket policies 279
Using bucket policies in Multicloud Object Gateway 279
Creating a user in the Multicloud Object Gateway 280
Bucket notification in Multicloud Object Gateway 281

Configuring bucket notification in Multicloud Object Gateway 281
Multicloud Object Gateway bucket replication 284
Replicating a bucket to another bucket 284
Replicating a bucket to another bucket using the MCG command-line interface 284
Replicating a bucket to another bucket using a YAML 285
Setting a bucket class replication policy 286
Setting a bucket class replication policy using the MCG command-line interface 286
Setting a bucket class replication policy using a YAML 287
Enabling log based bucket replication 287
Enabling log based bucket replication for new namespace buckets using OpenShift Web Console in AWS environment 288
Enabling log based bucket replication for existing namespace buckets using YAML 288
Enabling log based bucket replication in Microsoft Azure 289
Enabling log-based bucket replication deletion 290
Bucket logging for Multicloud Object Gateway 290
Enabling bucket logging by using the Best-Effort option 290
Enabling bucket logging by using the Guaranteed option 291
Synchronizing versions in Multicloud Object Gateway bucket replication 291
Obtaining metrics to reflect bucket replication state 292
Object Bucket Claim 293
Dynamic Object Bucket Claim 293
Creating an Object Bucket Claim using the command line interface 294
Creating an Object Bucket Claim using the OpenShift Web Console 296
Attaching an Object Bucket Claim to a deployment 296
Viewing object buckets using the OpenShift Web Console 297
Deleting Object Bucket Claims 297
Caching policy for object buckets 297
Creating an AWS cache bucket 297
Creating an IBM COS cache bucket 299
Lifecycle bucket configuration in Multicloud Object Gateway 300
Scaling Multicloud Object Gateway performance 300
Automatic scaling of MultiCloud Object Gateway endpoints 301
Autoscaling RGW in Fusion Data Foundation using HPA and KEDA 301
Scaling the Multicloud Object Gateway with storage nodes 302
Increasing CPU and memory for PV pool resources 302
Accessing the RADOS Object Gateway S3 endpoint 303
Using TLS certificates for applications accessing RGW 303
Accessing external RGW server 303
Using the security token service of Multicloud Object Gateway to assume the role of another user 304
Replacing nodes 304
Replacing nodes on Fusion Data Foundation using dynamic devices 305
Fusion Data Foundation deployed on AWS 305
Replacing an operational AWS node on user-provisioned infrastructure 305
Replacing an operational AWS node on installer-provisioned infrastructure 306
Replacing a failed AWS node on user-provisioned infrastructure 307
Replacing a failed AWS node on installer-provisioned infrastructure 308
Fusion Data Foundation deployed on VMware 308
Replacing an operational VMware node on user-provisioned infrastructure 309
Replacing an operational VMware node on installer-provisioned infrastructure 310
Replacing a failed VMware node on user-provisioned infrastructure 310
Replacing a failed VMware node on installer-provisioned infrastructure 311
Fusion Data Foundation deployed on Microsoft Azure 312
Replacing operational nodes on Azure installer-provisioned infrastructure 312
Replacing failed nodes on Azure installer-provisioned infrastructure 313
Fusion Data Foundation deployed on Google Cloud Platform 314
Replacing an operational node on Google Cloud Platform installer-provisioned infrastructure 314
Replacing a failed Google Cloud Platform node on installer-provisioned infrastructure 315
Replacing nodes on Fusion Data Foundation using local storage devices 315
Replacing devices 315
Dynamically provisioned Fusion Data Foundation deployed on AWS 316
Replacing an operational AWS node on user-provisioned infrastructure 305
Replacing an operational AWS node on installer-provisioned infrastructure 306
Replacing a failed AWS node on user-provisioned infrastructure 307
Replacing a failed AWS node on installer-provisioned infrastructure 308
Dynamically provisioned Fusion Data Foundation deployed on VMware 319
Replacing operational or failed storage devices on VMware infrastructure 319
Dynamically provisioned Fusion Data Foundation deployed on Azure 322
Replacing operational nodes on Azure installer-provisioned infrastructure 312
Replacing failed nodes on Azure installer-provisioned infrastructure 313
Dynamically provisioned Fusion Data Foundation deployed on Google Cloud Platform 323
Fusion Data Foundation deployed using local storage devices 324
Monitoring Fusion Data Foundation 324
Cluster health 324

Verifying Fusion Data Foundation is healthy 324
Storage health levels and cluster state 324
Viewing Fusion Data Foundation infrastructure health 325
Multicluster storage health 326
Enabling multicluster dashboard on Hub cluster 326
Verifying multicluster storage health on hub cluster 327
Metrics 327
Metrics in the Block and File dashboard 327
Metrics in the Object dashboard 328
Pool metrics 329
Network File System metrics 330
Enabling metadata on RBD and CephFS volumes 331
Verify the metadata for RBD PVC 331
Verify the metadata for RBD clone 332
Verify the metadata for RBD snapshots 333
Verify the metadata for RBD Restore 333
Verify the metadata for CephFS PVC 334
Verify the metadata for CephFS clone 335
Verify the metadata for CephFS volume snapshot 335
Verify the metadata of CephFS Restore 336
Alerts 337
Setting up alerts 337
Fusion Data Foundation metrics 337
Fusion Data Foundation service install and upgrade issues 357
Troubleshooting 358
Downloading log files and diagnostic information using must-gather 359
Running must-gather in modular mode 360
Using ODF CLI command 360
Commonly required logs for troubleshooting 362
Adjusting verbosity level of logs 363
Overriding the cluster-wide default node selector for Fusion Data Foundation post deployment 364
Encryption token is deleted or expired 364
Troubleshooting alerts and errors in Fusion Data Foundation 364
Resolving alerts and errors 365
Resolving cluster health issues 369
MON_DISK_LOW 369
Resolving cluster alerts 369
CephClusterCriticallyFull 370
CephClusterErrorState 371
CephClusterNearFull 371
CephClusterReadOnly 371
CephClusterWarningState 372
CephDataRecoveryTakingTooLong 372
CephFSStaleSubvolume 373
CephMdsCpuUsageHigh 373
CephMdsMissingReplicas 373
CephMgrIsAbsent 374
CephMgrIsMissingReplicas 374
CephMonHighNumberOfLeaderChanges 375
CephMonQuorumAtRisk 376
CephMonQuorumLost 376
CephMonVersionMismatch 377
CephNodeDown 377
CephOSDCriticallyFull 378
CephOSDDiskNotResponding 378
CephOSDDiskUnavailable 379
CephOSDFlapping 379
CephOSDNearFull 380
CephOSDSlowOps 380
CephOSDVersionMismatch 380
CephPGRepairTakingTooLong 381
CephPoolFull 381
CephPoolNearFull 381
CephPoolQuotaBytesCriticallyExhausted 381
CephPoolQuotaBytesNearExhaustion 382
CephXattrSetLatency 382
OSDCPULoadHigh 382
PersistentVolumeUsageCritical 382
PersistentVolumeUsageNearFull 383
NooBaaSystemCapacityWarning85 383
NooBaaSystemCapacityWarning95 383
NooBaaSystemCapacityWarning100 384

Resolving NooBaa Bucket Error State 384
Resolving NooBaa Bucket Exceeding Quota State 384
Resolving NooBaa Bucket Capacity or Quota State 385
Recovering pods 385
Recovering from EBS volume detach 385
Enabling debug logs for rook-ceph-operator 385
Disabling debug logs for rook-ceph-operator 386
Resolving low Ceph monitor count alert 386
Troubleshooting unhealthy blocklisted nodes 386
Checking for Local Storage Operator deployments 386
Removing failed or unwanted Ceph Object Storage devices 387
Verifying Ceph cluster is healthy 387
Removing failed or unwanted Ceph OSDs in dynamically provisioned Red Hat OpenShift Data Foundation 388
Removing failed or unwanted Ceph OSDs provisioned using local storage devices 388
Troubleshooting the error cephosd:osd.0 is NOT ok to destroy while removing failed or unwanted Ceph OSDs 389
Troubleshooting and deleting remaining resources during Uninstall 390
Troubleshooting CephFS PVC creation in external mode 391
Restoring the monitor pods in Fusion Data Foundation 392
Restoring the Multicloud Object Gateway 395
Restoring ceph-monitor quorum in Fusion Data Foundation 396
Changing resources for the Fusion Data Foundation components 398
Changing the CPU and memory resources on the rook-ceph pods 398
Tuning the resources for the MCG 399
Disabling Multicloud Object Gateway external service after deploying OpenShift Data Foundation 399
Accessing odf-console with the ovs-multitenant plug-in by manually enabling global pod networking 400
Annotating encrypted RBD storage classes 226
Force deletion of storage in provider clusters 400
Configuring CPU and memory resources for FDF-managed pods 400
Unable to delete incorrect IBM Scale (CNSA) cluster after first-time configuration 401
Filesystem deletion does not remove internal IBM Storage Scale resources 402
Fusion Data Foundation service error scenarios 403
Fusion Data Foundation storage node failure 406
Disk replacement and OpenShift dedicated OSD failure recovery 409
Storage client status reporter pod is stuck in terminating state 414
Global Data Platform 415
Installing Global Data Platform 415
Global Data Platform service install and upgrade issues 417
Uninstalling Global Data Platform 419
Configuring storage 420
Configuring Global Data Platform service on infrastructure nodes 420
Setting up encryption for storage 422
Configuring encryption for Global Data Platform storage 422
Troubleshooting Global Data Platform issues 423
Remote IBM Storage Scale storage cluster in IBM Fusion 424
Connecting to remote IBM Storage Scale file systems 424
Backup & Restore 426
Backup & Restore overview 427
Backup and restore planning 427
How Backup & Restore service backups up application data 428
Backup & Restore service architecture 428
Hub and spoke model overview 430
Install Backup & Restore 431
Prerequisites 431
Backup & Restore hub 431
Backup & Restore spoke 434
Establishing connection between hub and spoke 437
Uninstalling Backup & Restore 438
Getting started 438
Quick start Backup & Restore service 438
Service protection 439
Backup & Restore of your workloads 439
Backup & Restore of your applications and virtual machines 439
Backup storage locations 440
Creating a secret 440
Adding a backup storage location 441
Using MCG or Noobaa as a backup storage location 442
Using IBM Storage Protect as a backup storage location 443
Managing backup storage location details 443
Backup policies 444
Creating backup policy 444
Managing backup policy 445
Setting up backup policies 445

Managing your application workloads 446
Managing the application details 447
Running on demand backups 449
Jobs 449
Restoring an application 450
Backup and restore as code 451
Backup & restore service from OpenShift Container Platform 456
Service protection 459
Configuring service backups 460
Recovering service backups 460
Self-service Backup & Restore 461
Managing backup for multi-cluster environment 463
Connecting and managing cluster connections 464
Reconnecting OpenShift Container Platform cluster 465
Disabling the connection 466
Backup consistencies 466
Achieving data consistency for containers 466
Achieving data consistency for virtual machine workloads 467
Orchestrate a backup or restore 468
Custom backup and restore workflows 468
Creating a Recipe 469
Additional hooks 472
Dynamic recipe 474
Assigning a Recipe 477
Resource Transformation 479
Backup and restore of IBM Software Hub to the same or different cluster 483
IBM Fusion repository for recipes 483
Performance and scaling 483
Backup & restore resource self-monitoring 483
Backup & restore hub performance and scaling 484
Backup & restore configuration parameters 487
FIPS-140-2 494
Backup & Restore service installation and upgrade issues 494
Troubleshooting Backup & Restore service issues 496
Backup issues 497
Restore issues 500
IBM Software Hub backup and restore issues 505
Hub and spoke connection issues 506
Backup & restore service from OpenShift Container Platform console 507
Data Cataloging 507
IBM Data Cataloging architecture 508
Role-based access control 510
Data source connections 511
Cataloging metadata 511
Enriching metadata 511
Policy engine 512
Applications 512
Graphical user interface 512
Reports for IBM Data Cataloging 514
Installing IBM Data Cataloging 514
Uninstalling IBM Data Cataloging 518
Configuring IBM Data Cataloging 518
Setting up IBM Data Cataloging custom user and password 518
Configure data source connections 519
IBM Storage Scale data source connections 519
IBM Storage Scale scanning considerations 520
Security considerations 520
Performance considerations 520
Prerequisites for automated scanning 520
Identify the IBM Storage Scale cluster and node list 521
Creating or identifying a user ID and password for scanning 521
Validate scan user permissions and configuration 522
Check the dependencies for optimized scanning 522
Creating an IBM Storage Scale data source connection 522
Automated scanning of an IBM Storage Scale data source 524
Automated scanning of an IBM Storage Scale fileset 524
Manual scanning of an IBM Storage Scale data source 525
Performing file system scan to collect metadata from IBM Storage Scale 525
Copying the output of the IBM Storage Scale file system scan to the IBM Data Cataloging master node 527
Ingesting metadata from IBM Storage Scale file system scan in IBM Data Cataloging 527
Ingesting quota information from the file system 528
IBM Storage Archive data source connections 528

IBM Cloud Object Storage data source connection 528
Prerequisites 529
Creating an IBM Cloud Object Storage data source connection 529
Scanning an IBM Cloud Object Storage data connection 531
Best practices for scanning IBM Cloud Object Storage systems 533
Enabling bucket notifications for Ceph Object Storage 533
Enabling bucket notifications for Ceph Object Storage on Red Hat OpenShift 534
Replaying IBM Cloud Object Storage notifications 535
Overview of architecture 536
Configuration file 537
Replay performance 540
Replay tasks and vault settings 541
Including and excluding vaults 542
Stats files 543
Replay 543
Initialization for Replay 544
Error conditions 544
Output 544
Renaming a vault for Replay 544
Starting the Replay 545
Debug mode for Replay 545
Notifier 545
Limitations 545
Starting the Notifier 545
Notifier operation 546
Stopping the Notifier 546
Restarting the Notifier 546
Progress report 547
Logging 548
IBM Cloud Object Storage Scanner output data 549
Appendix 551
Configure IBM Cloud Object Storage notifications for IBM Data Cataloging 556
Enabling IBM Cloud Object Storage notification services 557
Authenticating, encrypting, and enabling 557
Authenticating, encrypting, and enabling services on Red Hat OpenShift 558
Testing the IBM Cloud Object Storage notification service 559
Monitoring the IBM Cloud Object Storage accesser logs 560
Monitoring the IBM Data Cataloging producer IBM Cloud Object Storage logs 560
Monitoring the IBM Data Cataloging dashboard for IBM Cloud Object Storage ingestion 560
IBM Data Cataloging and S3 object storage data source connections 561
Creating an S3 object storage data connection 561
Scanning an S3 object storage data connection 563
Best practices for scanning an S3 object storage system 565
Scanning an Elastic Storage Server data source connection 565
Creating a Network File System data source connection 566
Creating an SMB data source connection 566
Creating an IBM Storage Protect data source connection 567
Configuring data source connections offline 568
Editing and using the TimeSinceAccess and Size Range buckets 568
IBM Spectrum Storage software requirements 570
IBM Data Cataloging installation with alternative VLAN 571
Multiple connection managers 573
Enabling feature for skipping the snapshot directories 574
Enabling skip snapshot directories feature on Red Hat OpenShift 574
Managing IBM Data Cataloging rules when integrated with CAS 575
Administering IBM Data Cataloging 575
AI-enabled capabilities 576
IBM Data Cataloging MCP server 576
Content-based AI-enabled autotagging 580
Managing user access 583
Initial login 584
Resetting the sdadmin password 587
Password policies 587
Changing password 588
Managing user accounts 588
Creating user accounts 589
Managing groups 590
Creating groups 591
Managing collections 591
Creating collections 592
Managing LDAP and IBM Cloud Object Storage System connections 593
Creating an LDAP connection 593

Configuring self-signed certificates to secure LDAPS connection 594
Creating an IBM Cloud Object Storage connection 595
Managing metadata policies 596
Adding policies 596
Adding auto-tagging policy parameters 597
Adding deep-inspection policy parameters 597
Adding content search policy parameters 598
Running policies 598
Viewing policies 599
Viewing policy history details and logs 600
Modifying policies 600
Deleting policies 601
Using content search policies 601
Identifying the required regex expressions 602
Creating a content search policy 602
Viewing content search application logs 603
Hints and tips for using content search 603
Tiering data by using ScaleILM application 604
Viewing ScaleILM application logs 605
Copying data by using ScaleAFM application 606
Viewing ScaleAFM application logs 607
Exporting metadata to IBM Watson Knowledge Catalog 608
Configuring Watson Knowledge Catalog connector 608
Configuring Watson Knowledge Catalog using the utility script 609
Configuring Watson Knowledge Catalog manually 609
Exporting metadata 610
Mapping similar source connections in Watson Knowledge Catalog 610
Troubleshooting export issues 611
Authentication failure with Watson Knowledge Catalog - both on-Premise and IBM Cloud 611
Incorrect WKC URL configuration 611
Invalid connection type in catalog 612
No linked connection type 612
WKC connector pod in CrashLoopBackoff state 612
S3 connection issues 613
Importing externally curated tags for COS/S3 using import tags application 613
Initiating Policy Using REST API 614
Registering import tags application 614
Defining the import tag policy type 615
Viewing the policy status 615
Viewing the import tags application log 616
Performing retention analytics on IBM Storage Protect archive data 616
Enabling the retention analytics feature 617
Adding IBM Storage Protect data source and initiating scan 618
Searching expiry time 619
Managing tags 620
Creating tags 620
Viewing and searching tags 623
Editing tags 624
Deleting tags 626
Discover data 628
Searching 628
Grouping data by file type 629
Searching system and custom metadata fields 629
System metadata fields to search on 630
Access control list metadata to search on 631
How to use ACL data in policies 632
Search on custom metadata fields 634
Examples of search filters 635
Search results table 635
Refine search results 635
Sort search results 635
Tag search results manually 635
Managing applications 636
Reports 636
Creating a IBM Data Cataloging application for metadata-based policies 637
Graceful shutdown 638
Flushing Kafka topics 639
Returning IBM Data Cataloging to a running state 640
REST API for IBM Data Cataloging 642
IBM Data Cataloging REST APIs 642
API endpoints 643
Asynchronous endpoints 643

Status codes 644
Authentication process 644
Endpoints for working with a Db2 Warehouse 645
/db2whrest/v1/search: POST 645
/db2whrest/v1/report: POST 649
/db2whrest/v1/report: GET 655
/db2whrest/v1/report/<report_id>: GET 655
/db2whrest/v1/report/<report_id>/download: GET 656
/db2whrest/v1/bucket: GET 657
/db2whrest/v1/buckets/<bucket>: GET 657
/db2whrest/v1/report/<report_id>: PUT 658
/db2whrest/v1/buckets/<bucket>: PUT 658
/db2whrest/v1/report/<report_id>: DELETE 660
/db2whrest/v1/sql_query?<sql>: GET 660
/db2whrest/v1/sql_query: POST 661
/db2whrest/v1/sql_query_async?<sql>: GET 662
/db2whrest/v1/sql_query_async: POST 663
/db2whrest/v1/task_status/<task_id>: GET and <task_id>/peek 664
/db2whrest/v1/summary_tables -X GET 665
/db2whrest/v1/summary_tables/<table> -X GET 666
/db2whrest/v1/summary_tables/<table> -X PUT 666
/db2whrest/v1/summary_tables/<table>/<action> -X PUT 667
/db2whrest/v1/bulk_add_tags/docs: POST 668
Endpoints for working with policy management 668
/policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET 669
/policyengine/v1/policies/<policy_name>/preview: GET 670
/policyengine/v1/policies/<policy_name>/status:GET 671
/policyengine/v1/policyhistory: GET 672
/policyengine/v1/policyhistory/<pol_id>/<log_id>: GET 674
/policyengine/v1/policies/<policy_name>/<action>: POST 675
/policyengine/v1/policies -d '<data>': POST 675
/policyengine/v1/policies/<policy_name> -d '<data>': PUT 676
/policyengine/v1/policies/<policy_name>: DELETE 677
/policyengine/v1/policyhistory: DELETE 677
/policyengine/v1/tags: GET 678
/policyengine/v1/tags/: POST 679
/policyengine/v1/tags/: PUT 679
/policyengine/v1/tags/: DELETE 680
/policyengine/v1/regex: POST 680
/policyengine/v1/regex: GET 681
/policyengine/v1/regex/<regex_name>: GET 682
/policyengine/v1/regex: PUT 683
/policyengine/v1/regex: DELETE 684
Adding fields with the AUTOTAG action 684
Adding fields that contain initial values 684
Adding fields that are extracted by rule from existing fields 685
Endpoints for working with connection management 685
/connmgr/v1/connections: GET and /connmgr/v1/connections/<connection_name>: GET 686
/connmgr/v1/scan/<connection>/partitions: GET 687
/connmgr/v1/connections -d '<data>': POST 688
/connmanager/v1/scan/<connection>/partial: POST 690
/connmgr/v1/connections/<connection_name> -d '<data>': PUT 690
/connmgr/v1/connections/<connection_name>: DELETE 691
/connmgr/v1/scan/<connection_name>: POST 692
/connmgr/v1/scan/<connection_name>: GET 693
/connmgr/v1/scan: GET 693
/connmgr/v1/scan/<connection_name>: PUT 694
Application management by using APIs 695
/policyengine/v1/applications: POST 695
Example: Create a DeepInspect policy using the application 696
/policyengine/v1/applications/<application name>: GET 698
/policyengine/v1/tlscert: GET 698
/policyengine/v1/action_ids: GET 700
/policyengine/v1/applications/<deployment_name>/schema?action_id=<action_id>: GET 700
/policyengine/v1/application_names?action_id=<action_id>: GET 701
/policyengine/v1/applications: DELETE 702
RBAC management by using APIs 702
Managing tokens 703
/auth/v1/token: GET 703
Managing users 704
/auth/v1/users: GET 704
/auth/v1/users: POST 705

/auth/v1/users/<user_ID>/password: POST 706
/auth/v1/users/<user ID>: GET 707
/auth/v1/users/<user_ID>: PATCH 707
/auth/v1/users/<user_ID>: DELETE 708
/auth/v1/users/<user_ID>/groups: GET 708
/auth/v1/users/<user_ID>/collections: GET 709
/auth/v1/users/<user_ID>/roles: GET 710
/auth/v1/users/<user_ID>/role_assignment: GET 710
/auth/v1/users/users_summary: GET 711
Managing user roles 712
/auth/v1/roles/<role_ID>: PUT 712
/auth/v1/roles/<role_ID >: GET 714
/auth/v1/roles: GET 714
/auth/v1/roles/<role_ID>: DELETE 715
Managing user groups 716
/auth/v1/groups: GET 716
/auth/v1/groups: POST 717
/auth/v1/groups/<group ID>: GET 718
/auth/v1/groups/<group_ID>: PATCH 718
/auth/v1/groups/<group_ID>: DELETE 719
/auth/v1/groups/<group_ID>/collections: GET 719
/auth/v1/groups/<group_ID>/roles: GET 720
/auth/v1/groups/<group_id>/role_assignments: GET 720
/auth/v1/groups/<group_ID>/users: GET 721
/auth/v1/groups/<group_ID>/user/<user_ID>: DELETE 722
/auth/v1/groups/<group_ID>/user/<user_ID>: PUT 722
/auth/v1/groups/groups_summary: GET 723
Managing collections 724
/auth/v1/collections: GET 724
/auth/v1/collections: POST 725
/auth/v1/collections/<collection_ID>: GET 725
/auth/v1/collections/<collection_ID>: PATCH 726
/auth/v1/collections/<collection_ID>: DELETE 726
/auth/v1/collections/<collection_ID>/groups: GET 727
/auth/v1/collections/<collection_ID>/users: GET 728
/auth/v1/collections/<collection_id>/role_assignments: GET 728
Managing domains 729
/auth/v1/domains/<domain_ID>: GET 729
/auth/v1/domains: POST 730
/auth/v1/domains/<domain_ID>: PATCH 731
/auth/v1/domains/<domain_ID>: DELETE 731
Data protection 732
Backup and restore 732
Backup and restore scripts 732
Disaster recovery procedures 733
Preparations for disaster recovery 733
Running disaster recovery for Red Hat OpenShift 733
Using a third-party data movement application to manage data 733
Data movement with IBM Data Cataloging and Moonwalk 735
Preserving tags during data movement 735
Monitoring IBM Data Cataloging 736
Monitoring data sources 736
Viewing data source status 736
Viewing data source connections 740
Recommended to move 740
Deleting or editing a connection 741
Monitoring the IBM Data Cataloging service environment 743
Audit log 743
Configurable log trimmer handler 744
SD-Monitor 744
Health check monitoring 744
Collecting logs and metrics 747
Data Cataloging Harvester CLI 748
FKEY migration script 750
Configurable Db2 log trimmer 751
IBM Data Cataloging service installation and upgrade issues 752
IBM Fusion IBM Data Cataloging known issues 755
Content-Aware Storage (CAS) 760
Architecture 761
Understanding user personas 761
Installing Content-Aware Storage (CAS) 762
Planning and prerequisites 763

Installing CAS without a remotely mounted IBM Storage Scale filesystem 765
Installing NVIDIA NeMo Retriever Library 765
Configuring NVIDIA Inference Microservices (NIMs) 767
Configuring a Multi-Instance GPU (MIG) with CAS 768
Configuring IBM Storage Scale user to enable watch creation 770
Configuring Content-Aware Storage (CAS) 770
Integrating CAS and IBM Data Cataloging 771
Integrating IBM Data Cataloging for file filtering 771
Using IBM Data Cataloging attributes with CAS 772
IBM Data Cataloging and Content-Aware Storage insights 776
Creating a data source 776
Prerequisites for S3 and NFS type data sources 778
Deploying an S3 data source with custom SSL certificate 778
Creating a domain and connecting with data source 779
Configuring Document Processor Engine 780
Supported document types 780
IBM Docling Multimodal processing engine 780
Configuring Docling Multimodal GPUs 781
NVIDIA Multimodal processing engine settings 781
Configuring Content-Aware Storage (CAS) through CLI 782
Using your own Hugging Face access token with CAS for NVIDIA Multimodal 785
Configuring CAS Query Access Control and Security 785
Configuring CAS Resource Access Control (CRAC) 786
Configuring CAS to connect to your Identity Provider (IDP) 787
Configuring file-level access control in CAS 788
Client application with Content-Aware Storage (CAS) 790
Content-Aware Storage (CAS) APIs 791
Integrating Model Context Protocol (MCP) with CAS 795
Integrating CAS with watsonx Orchestrate by using MCP 798
Monitoring 801
Uninstalling Content-Aware Storage (CAS) service 802
Cleaning up Content-Aware Storage (CAS) data 803
Troubleshooting issues and known limitations in Content-Aware Storage (CAS) 803
Related information 815
Lakehouse-Table [Technology preview] 815
Installing Lakehouse-Table service 816
Configuring Lakehouse-Table service 818
Troubleshooting issues and known limitations in Lakehouse-Table 820
Deploying IBM Fusion services using CLI 822
Disaster recovery 824
Disaster recovery configuration for Fusion Data Foundation 825
Metro-DR configuration for Fusion Data Foundation 825
Components of Metro-DR solution 826
Unsupported features for Metro-DR 827
Metro-DR deployment workflow 827
Requirements for enabling Metro-DR 828
Requirements for deploying IBM Storage Ceph stretch cluster with arbiter 828
Hardware requirements 828
Software requirements 829
Network configuration requirements 829
Deploying IBM Storage Ceph 829
Node pre-deployment steps 829
Cluster bootstrapping and service deployment with Cephadm 831
Configuring IBM Storage Ceph stretch mode 834
Installing Fusion Data Foundation on managed clusters 836
Installing Fusion Data Foundation Multicluster Orchestrator operator 837
Configuring SSL access across clusters 838
Creating Disaster Recovery Policy on Hub cluster 839
Configure DRClusters for fencing automation 840
Add node IP addresses to DRClusters 840
Add fencing annotations to DRClusters 841
Create sample application for testing disaster recovery solution 841
ApplicationSet-based applications
Creating ApplicationSet-based applications 841
Apply Data policy to sample ApplicationSet-based application 842
Deleting a sample application 842
Subscription-based application failover between managed clusters 843
ApplicationSet-based application failover between managed clusters 844
Relocating Subscription-based application between managed clusters 844
Relocating an ApplicationSet-based application between managed clusters 845
Recovering to a replacement cluster with Metro-DR 846
Hub recovery using Red Hat Advanced Cluster Management [Technology preview] 849

Configuring passive hub cluster 849
Switching to passive hub cluster 849
Regional-DR configuration for Fusion Data Foundation 850
Components of Regional-DR solution 851
Hybrid cloud Regional-DR solution 852
Regional-DR deployment workflow 852
Requirements for enabling Regional-DR 852
Creating a Fusion Data Foundation cluster on managed clusters 854
Installing Fusion Data Foundation Multicluster Orchestrator operator 837
Configuring SSL access across clusters 838
Creating Disaster Recovery Policy on Hub cluster 856
Disaster recovery protection for cloned and restored RBD volumes 857
Create sample application for testing disaster recovery solution 857
ApplicationSet-based applications
Creating an ApplicationSet-based application 858
Applying DRpolicy to an ApplicationSet-based application 858
Deleting a sample application 842
Virtual machine garbage collection automation 860
Subscription-based application failover between managed clusters 861
ApplicationSet-based application failover between managed clusters 861
Relocating a Subscription-based application between managed clusters 862
Relocating an ApplicationSet-based application between managed clusters 863
Disaster recovery protection for discovered applications 864
Prerequisites for disaster recovery protection of discovered applications 864
Creating a sample discovered application 865
Enrolling a sample discovered application for disaster recovery protection 866
Discovered application failover and relocate 867
Failover disaster recovery protected discovered application 868
Relocate disaster recovery protected discovered application 869
Disable disaster recovery for protected applications 869
Recovering to a replacement cluster with Regional-DR 870
Viewing Recovery Point Objective values for disaster recovery enabled applications 873
Hub recovery using Red Hat Advanced Cluster Management 873
Configuring passive hub cluster 873
Switching to passive hub cluster 874
Enabling granular disaster recovery for individual or groups of virtual machines in a namespace [Technology preview] 875
Disaster recovery with stretch cluster for Fusion Data Foundation 876
Requirements for enabling stretch cluster 876
Applying topology zone labels to OpenShift Container Platform nodes 877
Installing Local Storage Operator 877
Installing IBM Fusion Data Foundation Operator 878
Creating Fusion Data Foundation cluster 878
Verifying Fusion Data Foundation deployment 880
Verifying the state of the pods 880
Verifying the Fusion Data Foundation cluster is healthy 881
Verifying the Multicloud Object Gateway is healthy 881
Verifying that the specific storage classes exist 882
Installing zone-aware sample application 882
Scaling the application after installation 883
Modifying deployment to be zone-aware 884
Recovering Fusion Data Foundation stretch cluster 885
Understanding zone failure 886
Recovering zone-aware HA applications with rwx storage 886
Recovering HA applications with rwx storage 886
Recovering applications with RWO storage 886
Recovering StatefulSet pods 887
Monitoring disaster recovery health 887
Enable monitoring for disaster recovery 888
Enabling disaster recovery dashboard on Hub cluster 888
Viewing health status of disaster recovery replication relationships 889
Viewing health status of ApplicationSet-based and Subscription-based applications 889
Disaster recovery metrics 889
Disaster recovery alerts 890
Troubleshooting disaster recovery 892
Troubleshooting Metro-DR 892
A StatefulSet application stuck after failover 892
DR policies protect all applications in the same namespace 892
During failback of an application stuck in Relocating state 893
Relocate or failback might be stuck in Initiating state 893
Troubleshooting Regional-DR 893
rbd-mirror daemon health is in warning state 893
volsync-rsync-src pod is in error state as it is unable to resolve the destination hostname 894

Cleanup and data sync for ApplicationSet workloads remain stuck after older primary managed cluster is recovered post failover 894
VM recipe known issues 895
Troubleshooting 2-site stretch cluster with Arbiter 895
Recovering workload pods stuck in ContainerCreating state post zone recovery 895
Disabling disaster recovery for a disaster recovery-enabled application 896
Disabling disaster recovery for managed applications 896
Disabling disaster recovery for discovered applications 896
Uninstalling disaster recovery from hub and managed clusters 896
Prerequisites for disaster recovery uninstallation 897
Uninstalling disaster recovery for peer-managed clusters 897
Uninstalling disaster recovery operators 898
Multi-cluster IBM Fusion using Hosted Control Plane 898
Interoperability matrix 899
Prerequisites for Hosted Control Plane clusters 899
Configuring a Hosted Control Plane cluster in a proxy environment 902
Deployment of virtualized clusters with Fusion Data Foundation 903
Planning and prerequisites for your virtualized clusters 903
Creating a virtualized Hosted Control Plane cluster 903
Deploying Bare Metal Hosted Control Plane cluster in disconnected environment 904
Deploying virtualized Hosted Control Plane cluster in disconnected environment 908
Deployment of Bare Metal clusters with Fusion Data Foundation 910
Planning and prerequisites for your Bare Metal Hosted Control Plane 911
Deploying Bare Metal clusters with Fusion Data Foundation 916
Configuring ingress for Bare Metal Hosted cluster 917
Expanding your Hosted Control Plane Bare Metal cluster 918
Installing the IBM Fusion base 919
Installation of data services on a Hosted Control Plane cluster 919
Installing Fusion Data Foundation on a Hosted Control Plane cluster 919
Deploying storage client in the Hosted Control Plane clusters 920
Installing Global Data Platform remote mount on the hosted cluster 922
Backup & Restore of applications in Hosted Control Plane clusters 922
Installing Backup & Restore on the hosted cluster 922
Updating the storage quota of a Hosted Control Plane cluster 923
Cleaning up of a Hosted Control Plane cluster 924
Troubleshooting issues and known limitations in Hosted Control Plane cluster 925
Red Hat OpenShift Virtualization 927
Overview 927
Red Hat OpenShift Virtualization Sizing on IBM Fusion 927
Sizing calculator 928
Setting storage for virtual machines 930
Deploying Red Hat OpenShift Virtualization Operator 930
Configuring GPU access with NVIDIA vGPU 930
Migrating VMs to OpenShift Virtualization 931
Migrating VMs from VMware vSphere to OpenShift 931
High-availability of virtual machines 931
Live migration for virtual machines 932
Virtual machine failover 933
Backup and restore of virtual machines 934
Creating a virtual machine by using Clone PVC 935
Disaster recovery for virtual machines 935
Advanced workloads 936
IBM Cloud Paks support for IBM Fusion 936
Validation tools for IBM Cloud Paks 937
IBM Maximo® Applications support for IBM Fusion 938
IBM Maximo® Application Suite 939
IBM Maximo® on Hosted Control Plane 940
IBM watsonx 940
AI workloads on IBM Fusion 941
Administering 941
Observability 941
AI-enabled observability with Red Hat OpenShift Lightspeed 941
IBM Fusion metrics 942
Enabling observability for IBM Fusion clusters 942
IBM Fusion Call Home Telemetry 943
Knowing your IBM Fusion user interface 943
Working with the dashboard 944
Understanding the IBM Fusion license report 944
Browser requirements 945
Icons used in the user interface 945
IBM Fusion user interface issues 946
Upgrading IBM Fusion 946
Prerequisites for enterprise registry upgrade 947
Upgrading IBM Fusion 948
Enhanced upgrade experience 949
Upgrading Hosted Control Plane Fusion and its services 950

Upgrading OpenShift Container Platform on Hosted Control Plane 950
Upgrading OpenShift Container Platform 950
Upgrading IBM Fusion services 951
Migrating Global Data Platform 951
Upgrading Global Data Platform 952
Upgrading Fusion Data Foundation 953
Upgrading Red Hat OpenShift Data Foundation to IBM Storage Fusion Data Foundation 954
Upgrade Data cataloging 955
Upgrading Backup & Restore service 955
Upgrading Content-Aware Storage (CAS) 956
Troubleshooting and serviceability 957
Contacting IBM Support Center 957
How to provide feedback 958
Serviceability in IBM Fusion 958
Analyzing events in IBM Fusion 958
IBM Fusion event alerts 959
Collecting logs in IBM Fusion 960
Log package 961
Automatic cleanup of log packages 962
Benefits of enabling Call Home 963
Data privacy with Call Home 964
Enabling Call Home 964
Configuring Call Home with proxy 966
IBM Fusion Call Home Telemetry 943
Custom Certificate Authority (CA) support for Call Home 967
Configuring License Service Reporter for IBM Fusion 967
Events and error codes message references in IBM Fusion 969
Install events and error codes 970
BMYIN1501 971
BMYIN1502 971
BMYIN1503 972
BMYIN1504 972
BMYIN1505 972
BMYIN1506 972
BMYIN1507 972
BMYIN1508 973
BMYIN1509 973
BMYIN1510 973
BMYIN1512 973
BMYIN1513 973
BMYIN1514 974
BMYIN2501 974
BMYIN2502 974
BMYIN2503 974
BMYIN3116 974
BMYIN3117 975
BMYIN3119 975
BMYIN4101 975
BMYIN4102 976
BMYIN4103 976
BMYIN4104 976
BMYIN4105 976
BMYIN4106 977
BMYIN4107 977
BMYIN4108 977
BMYIN4110 977
BMYPC5010 978
BMYPC5027 978
BMYPC6006 978
BMYPC6007 979
BMYPC6008 980
BMYPC6010 980
BMYPC6013 981
BMYPC6040 981
BMYPC6042 982
BMYPC6044 982
BMYPC6045 982
Upgrade events and error codes 983
BMYUP1101 983
BMYUP1102 983
BMYUP1103 984
BMYUP1104 984

BMYUP1111 984
BMYUP1112 984
BMYUP1113 984
BMYUP1114 985
BMYUP1115 985
BMYUP1116 985
BMYUP1117 985
BMYUP1118 985
BMYUP1119 986
BMYUP1120 986
BMYUP1121 986
BMYUP1123 986
BMYUP3101 986
BMYUP3102 987
BMYUP3103 987
BMYUP3104 987
BMYUP3105 987
BMYUP3106 988
BMYPC6021 988
BMYPC6022 988
BMYPC6053 989
Global Data Platform events and error codes 989
BMYSS0001 989
BMYSS0002 989
BMYSS0003 990
BMYSS0004 990
BMYSS0005 990
BMYSS0006 990
BMYSS0007 991
BMYSS0009 991
BMYSS0010 991
BMYSS0011 991
BMYSS0012 992
BMYSS0013 992
Data foundation events and error codes 992
BMYSS0100 993
BMYSS0101 993
BMYSS0102 993
BMYSS0103 994
BMYSS0104 994
BMYSS0105 994
BMYSS0106 994
BMYSS0107 994
BMYSS0108 995
BMYSS0109 995
BMYSS0110 995
BMYSS0111 995
BMYSS0112 996
BMYSS0115 996
BMYSS0116 996
BMYSS0117 996
BMYSS0118 996
BMYSS0119 997
BMYSS0120 997
BMYSS0121 997
BMYSS0122 997
BMYSS0125 998
BMYSS0126 998
BMYSS0127 998
BMYSS0128 998
IBM Data Cataloging events and error codes 999
BMYDC0001 999
BMYDC0002 999
BMYDC0003 1000
BMYDC0004 1000
BMYDC0005 1001
BMYDC0006 1001
BMYDC0100 1001
BMYDC0101 1001
BMYDC0102 1002
BMYDC0103 1002
BMYDC0104 1002

BMYDC0105 1002
BMYDC0106 1003
BMYDC0107 1003
BMYDC0108 1003
BMYDC0109 1003
BMYDC0110 1003
BMYDC0113 1004
BMYDC0114 1004
BMYDC0115 1004
BMYDC0116 1004
BMYDC0117 1005
BMYDC0118 1005
Backup & Restore events and error codes 1005
BMYBR0001 1005
BMYBR0002 1006
BMYBR0003 1006
BMYBR0009 1006
BMYDP1146 1006
BMYDP1163 1007
Connection services events and error codes 1007
BMYCS100 1007
BMYCS101 1007
BMYCS102 1008
BMYCS103 1008
BMYCS300 1008
BMYCS301 1008
BMYCS302 1008
BMYCS303 1009
BMYCS304 1009
BMYCS305 1009
BMYCS306 1009
Content-Aware Storage events and error codes 1010
BMYCA0201 1010
BMYCA0204 1011
BMYCA0216 1011
BMYCA0217 1011
BMYCA0225 1011
BMYCA0227 1012
BMYCA0228 1012
BMYCA0229 1012
BMYCA0230 1013
BMYCA0231 1013
BMYCA0232 1013
BMYCA0233 1014
BMYCA0234 1014
BMYCA0235 1014
BMYCA0243 1015
BMYCA0244 1015
BMYCA0245 1015
BMYCA0258 1016
BMYCA0259 1016
BMYCA0260 1016
BMYCA0261 1017
BMYCA0263 1017
BMYCA0264 1017
BMYCA0266 1018
BMYCA0267 1018
BMYCA0268 1018
Serviceability events and error codes 1019
BMYLC0001 1019
BMYLC0002 1019
Troubleshooting issues in IBM Fusion 1019
Connection setup after OpenShift Container Platform cluster recovery 1020
Network policies 1020
Virtualization issues 1021
Frequently asked questions 1021
Install 1022
IBM Fusion Access for SAN 1022
Data Foundation Storage 1025
Backup and Restore 1025
IBM Data Cataloging 1026
Content-Aware Storage (CAS) 1026

Support and serviceability 1028

Accessibility features for IBM Fusion 1030

IBM Fusion considerations for GDPR readiness 1030

Glossary 1032
A 1032
B 1033
C 1033
D 1034
E 1035
F 1035
G 1036
H 1036
I 1036
J 1037
K 1037
L 1037
M 1037
N 1037
O 1038
P 1038
Q 1038
R 1038
S 1039
T 1040
U 1041
V 1041
W 1041
X 1041
Z 1042

Notices 1042
EU Data Act (Regulation 2023/2854) 1044

Product overview

IBM Fusion is a container-native hybrid cloud data platform that offers simplified deployment and data management for Kubernetes applications on Red Hat® OpenShift® Container Platform. IBM Fusion is designed to meet the storage requirements of modern, stateful Kubernetes applications and to make it easy to deploy and manage container-native applications and their data on Red Hat OpenShift Container Platform.

IBM Fusion is an advanced storage and backup solution that is designed to simplify data accessibility and availability across hybrid clouds. Companies can expand data availability across complex hybrid clouds for greater business performance and resilience. With the IBM Fusion solutions, organizations manage only a single copy of data. They need not create duplicate data when moving application workloads across the enterprise, easing management functions when you streamline analytics and AI.

IBM Fusion provides a streamlined way for organizations to discover, secure, protect, and manage data from the edge, to the core data center, to the public cloud.

IBM Fusion is available as two deployments, namely IBM Fusion HCI and IBM Fusion.

IBM Fusion HCI

IBM Fusion HCI is a purpose-built, hyper-converged architecture that is designed to deploy bare metal Red Hat OpenShift container management and deployment software alongside IBM Fusion software. For consistent and rapid deployment and management, it features an appliance form-factor, hyper-converged infrastructure along with integrated software-defined storage to meet the storage requirements of modern, stateful Kubernetes applications. Built with a storage platform that includes the essential elements necessary for mission-critical containers and hybrid cloud, the IBM Fusion provides a comprehensive infrastructure with compute, networking, and storage resources, including a data platform and global data services for Red Hat OpenShift.
The following *Figure 1. High-level conceptual overview of IBM Fusion HCI* provides a high-level conceptual overview of IBM Fusion HCI architecture:

- The IBM Fusion HCI provides a fully integrated infrastructure, which combines compute, networking, and storage resources optimized for containerized workloads.
- Designed to run on Bare Metal Red Hat OpenShift Container Platform, it supports the building and running of cloud-native applications.
- The Fusion Base offers essential services such as serviceability for issue diagnosis and resolution, a user interface for interaction with IBM Fusion HCI, and observability for monitoring system performance and health.
- The Fusion infrastructure management layer, built on top of OpenShift, simplifies lifecycle management of compute nodes, networking, and storage, ensuring operational efficiency.
- The Fusion services layer delivers a suite of solutions to secure and manage data across hybrid cloud environments.

Figure 1. High-level conceptual overview of IBM Fusion HCI



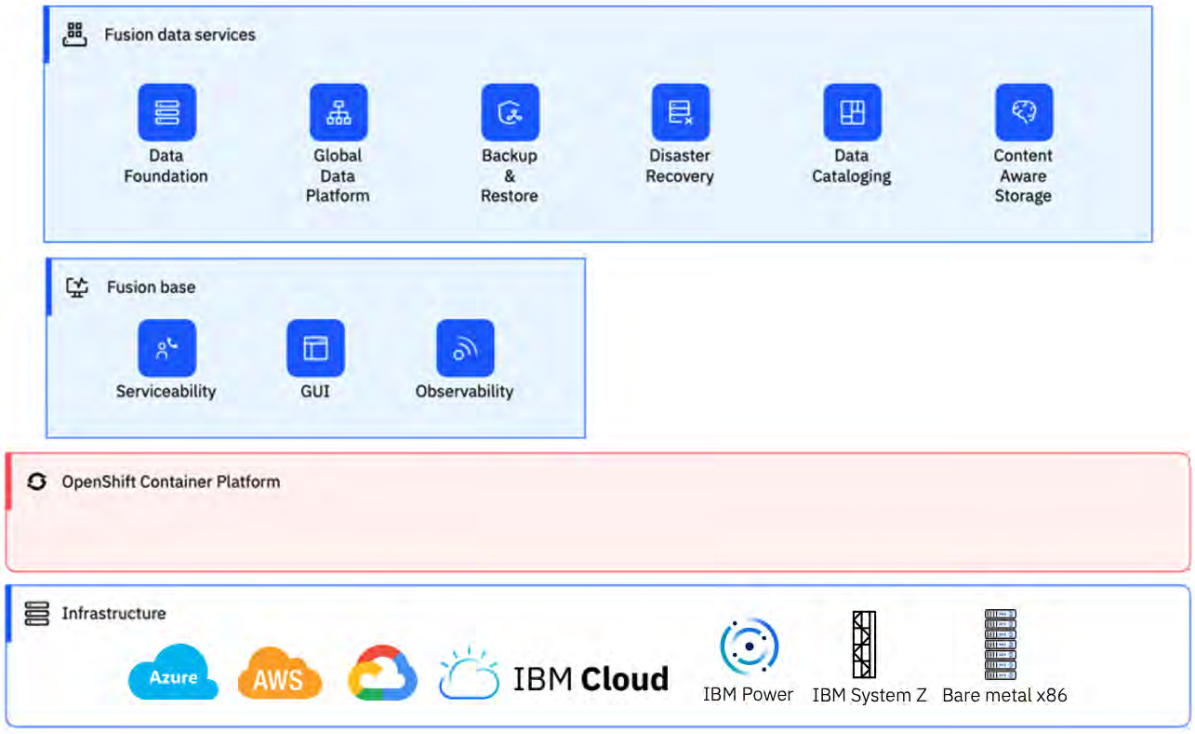
For more information about IBM Fusion HCI, see IBM Fusion HCI documentation.

IBM Fusion

IBM Fusion is software-defined storage with protection, backup, and caching elements and can be run on existing hardware resources.
The following *Figure 2. High-level conceptual overview of IBM Fusion* provides a high-level conceptual architecture overview:

- IBM Fusion is designed to run on OpenShift Container Platform across your infrastructure and various supported platforms, enabling the building and running of cloud-native applications.
- Built atop OpenShift, Fusion Base provides foundational services such as serviceability for efficient issue diagnosis and resolution, a user interface for seamless interaction with IBM Fusion, and observability for system performance insights.
- Layered above Fusion Base, Fusion services deliver a suite of solutions to secure and manage data across hybrid cloud environments.

Figure 2. High-level conceptual overview of IBM Fusion



- **Benefits of IBM Fusion**
  Benefits offered by IBM Fusion.
- **IBM Storage Expert Care**
  IBM Storage Expert Care is a simplified method of selecting services and support for systems at the time of the purchase. IBM Storage Expert Care is designed to simplify and standardize the support approach with transparent pricing and selection of services.
- **IBM Fusion trial version**
  IBM offers a 60-day free trial to experience IBM Fusion on any platform.
- **Fusion Essentials**
  Fusion Essentials refers to the restricted use of IBM Fusion that is defined in the license terms of a growing list of IBM Software products.
- **Security in IBM Fusion**
  IBM Fusion is a secure platform to deploy your applications.
- **Verifying image signatures**
  Digital signatures provide a way for consumers of content to ensure that what they download is both authentic (it originated from the expected source) and has integrity (it is what we expect it to be). All images for IBM Fusion are signed. This page describes how to verify the signatures on those images.

Benefits of IBM Fusion

Benefits offered by IBM Fusion.

- Modernize AI workloads
  Supports GPU accelerated applications with NVIDIA A100 GPU nodes and integrates IBM Cloud Paks to infuse AI. For more information about the supported versions of IBM Cloud Paks and prerequisites, see IBM Cloud Paks support for IBM Fusion.
- Organize and optimize resources
  Get file, volume, and object access across disparate data sources — including traditional storage, IBM storage, cloud, and edge.
- Enterprise storage and data services
  Publish storage profiles and protection policies as automated deployment models designed for seamless use.
- Access data without data movement
  Provide data accessibility both universally across platforms and locally across the information estate.
- Automate data protection
  Provision data using policies and protection methods stretching from local disk to remote to cloud and even to tape.
- Adjust resources non-disruptively
  Shift data from one repository type to another without the need to refactor the data.
- Adopt new services non-disruptively
  Roll any IBM Fusion updates into the data services automatically and without intervention.
- Hybrid cloud integration
  Improve application collaboration that is connecting data from the data center to the cloud and bringing more application agility. It integrates the data center to public cloud resources.

IBM Storage Expert Care

IBM Storage Expert Care is a simplified method of selecting services and support for systems at the time of the purchase. IBM Storage Expert Care is designed to simplify and standardize the support approach with transparent pricing and selection of services.

For more information about IBM Storage Expert Care for IBM Fusion, see Support reference guide.

IBM Fusion trial version

IBM offers a 60-day free trial to experience IBM Fusion on any platform.

Follow the steps to quickly access the IBM Fusion trial version:

- Register for a IBM Fusion 60-day trial.
- Obtain your IBM entitlement key from the container software library in My IBM.
- Deploy IBM Fusion per the following documentation for the target platform.
- Connect with the IBM Fusion Community. Get expert advice, share experiences, and collaborate to solve key challenges.

Fusion Essentials

Fusion Essentials refers to the restricted use of IBM Fusion that is defined in the license terms of a growing list of IBM Software products.

The restricted use terms entitles the user to deploy the Fusion Data Foundation service of IBM Fusion within the same cluster as the licensed IBM Software product.

Fusion Essentials allows up to 12 TB of capacity for use by the licensed IBM Software product. It is fully supported by IBM with support requests initiated through the licensed IBM Software product.

IBM Fusion, included with IBM Software, allows application owners to get deployed immediately while having confidence to add additional capacity and/or capabilities required as needs grow to large-scale mission-critical deployments.

Included with Fusion Essentials:

- Fusion Data Foundation internal mode deployment
- Up to 12 TB of usable capacity
- Compression
- Cluster-wide encryption
- Cross-availability zone HA
- Fully supported by IBM

For additional functionality, users can purchase a full IBM Fusion Advanced license.

Require Fusion Advanced license:

- Capacity requirements of more than 12 TB per cluster
- External mode deployment
- Backup
- Disaster recovery
- Global Data Platform (IBM Storage Scale)
- Data catalog
- Advanced encryption with external key management
- Usage with applications beyond the licensed IBM software

Get started with Fusion Essentials:

1. Deploy IBM Fusion Operator in your cluster. For more information about the installation, see Deploying IBM Fusion.
2. Deploy and configure the Fusion Data Foundation service in your cluster. For more information about the Fusion Data Foundation installation, see Installing Fusion Data Foundation.

Security in IBM Fusion

IBM Fusion is a secure platform to deploy your applications.

The following security measures are followed for IBM Fusion product:

- Penetration testing to check for exploitable vulnerabilities
- Threat modelling and threat assessment to ensure that threats are assessed and resolved
- Static scan to maintain high code quality and to ensure that no security vulnerabilities are left in the code
- Dynamic scan to detect and remove runtime vulnerabilities
- Open Source Scan to detect and remove open source vulnerabilities in open source libraries that are used by IBM Fusion
- Security and Privacy by Design (SPbD) compliance certified by IBM BISO to ensure security and privacy compliance
- Container Software certifications (IBM Certification and Red Hat Certifications) to ensure adherence to container security standards.
- Boundary protection: To enforce strict network security and boundary protection, IBM Fusion implements comprehensive ingress and egress Kubernetes `NetworkPolicies`. These policies operate on a deny-by-default, least-privilege architecture. Only explicitly required communication paths between specific

namespaces and pods are permitted, which drastically reduces the attack surface and prevents unauthorized lateral movement within the cluster.
- IBM Fusion uses Federal Information Processing Standards (FIPS)-certified cryptographic modules to ensure secure operations in regulated environments. For more information, see IBM Fusion FIPS capabilities (Optional).

Verifying image signatures

Digital signatures provide a way for consumers of content to ensure that what they download is both authentic (it originated from the expected source) and has integrity (it is what we expect it to be). All images for IBM Fusion are signed. This page describes how to verify the signatures on those images.

Before you begin

Your machine must have these command line tools installed (they can usually be installed on Linux using the package manager):

- GNU Privacy Guard v2
- OpenSSL
- skopeo

The IBM Fusion public key must exist on the same machine. Copy the following into a text editor, and save it in a file named `storage-fusion.pub.asc`:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGPOH0IBEADA253RAyBkEFFO7d9gfu+hdG58qQUe2H+4m5MuLc0zy/ulhkGg
YGBWFBCutkMp9F/jvtTrhf8r0oSlrIIzyJHFpM6qnlLftkbzQlpaj8yT+aPpNgVm
1xF5+hS32ouPXlUfaP3sX5CL0VZ44UhL0FBszfps7EPQU4C9fp6Rwstgejt+k5Xc
LIGiUIa+Kr3bmrBqN8nhsTE5csMlzS6tAIiG1R1I3qGRVcICpPh3lN7Mw2PYH8mB
kr0+a5pWQLJa/XSac9VHoDxoLCY8aPuxfLZUbTKEVkuGMXWvBHPVNYScydos5Jd9
Lzpc5uP6a8+7SBdMGZ5rupVkDKAZRTn+bxvI49D1yO0DBbXLaOMYevXUCIDeq3DI
X5BLNUZy8j6283IEReBTol8HnefpCT4mXQijFCIk595JiZwbcCY23XDdooXeTwk1
WZQI/TeQdV2q+gSS+tmdj3lD/iolXbNXneBK1G858sgct2hTIqZ/S2B3R94lDL2N
neV4IcsRy4r00bG2OdyLX/24yV857U427s1X56zc740pYgF/Uu61dauviXyaU2UD
C2KBy/YWmpWaMAeqNLRRYcHCUb9HZCfWchPlywXWIIggjL3D8vrPePWITpblC4wF
Tl6utwKHKM1CUqjn20j6q2Mh/7Z7g5XwaqmtoDm8cpx1bRY2ChHHxPXzbQARAQAB
tCpJQk0gU3BlY3RydW0gRnVzaW9uIEhDSSA8cHNpcnRAdXMuaWJtLmNvbT6JAjoE
EwEIACQFAmPOH0ICGw8FCwkIBwIGFQoJCAsCBBYCAwECHgEFCQPCZv8ACgkQHS4H
jVltgQNUmQ/+J9xUtUs1PwzAB/LpoqHGOmyZqz6EmQbaRiPzD4VSxcTYbpSKVb07
5H1sVR0QgDR9OUt5cbc4d9NBtFGiY8UeiCSL+MPjy5ksjFmbsxLRKEYZjB/qiOHv
c8viMH4y0boquomjeqGL1qajKyUvG1h99S5osGJVmPa2tmxMhTai57+aDX5SAKS4
RO1fd52K4nZw5+iqy95fXoEkPNwdw6Xm2fZ10F+kZ2dS8y+gu45hEaa/h+3P2myJ
37mp+336cfrqmqIv0pXD64et8zsQRNYqr2QEUSNsJ7ok4RkRqlpku5RYTV7T7TqH
g88rLAY1vxw4RzkQMve+t1IqgLANG9A+/fbQUQ/yshjCmvqmzB54VrOjOIwMWVtL
J+u1jbrdX1kFPogei6c0v3CBNe1khO6+1amBPTazKONbuMhBvaseX16x4Mk5r0FS
4FLNDEjktk4qUQAkEZk/Sugozp6oQpMYmKxPIt/7snLL9e6naF98TosKrpUCkme3
phbRIsML7Z7rUWg+jGS0Iz2L4FnVGrFs8iKsq23jvAf02nTKPtfgbozp6wMRC/KL
Wn3d9bGePGCMwPNwn3Auo8w/qMvHNAHykwTq1T1r/1vrmOnx+EEar2J2ZXqoA82P
4hcXWG4auoSA3y9tjAVLWChZdawFcMqcHZlHxER9/GOnY3hP+WSmv8o=
=YUDL
-----END PGP PUBLIC KEY BLOCK-----
```

You must have a list of images to verify. To get a list of container images used, see the procedure in Installing IBM Fusion from enterprise registry.

In this procedure, the following example image is used: `icr.io/cpopen/isf-operator-catalog:2.13.0-linux.amd64`.

Procedure

1. Run the following command to delete all previous IBM Fusion public keys:

   ```
   sudo gpg2 --delete-key "Fusion"
   ```

   To confirm the key deletion, enter 'Y'. If you have previous IBM Fusion public keys, repeat this step until gpg2 reports:

   ```
   gpg: key "Fusion" not found: Not found
   gpg: Fusion: delete key failed: Not found
   ```

2. Import the IBM Fusion public key on the machine that you prepared according to the *Before you begin* section:

   ```
   sudo gpg2 --import storage-fusion.pub.asc
   ```

   Note: This step must be done only once on each machine you use for signature verification.
3. Calculate the fingerprint:

   ```
   fingerprint=$(sudo gpg2 --fingerprint --with-colons "Fusion" | grep fpr | tr -d 'fpr:')
   ```

   This command stores the key's fingerprint in an environment variable that is called `fingerprint`, which is needed for the command to verify the signature. When you exit your shell session, the variable gets deleted. The next time that you log in to your machine, you can set it again by rerunning the command.

4. Create a directory for the image and use `skopeo` to pull it into local storage:

   ```
   mkdir images
   FUSION_VERSION=<version of Fusion being installed/verified>
   skopeo copy docker://icr.io/cpopen/isf-operator-catalog:${FUSION_VERSION}-linux.amd64 dir:./images
   ```

   For example, to check the signatures on 2.13.0:

```
mkdir images
FUSION_VERSION=2.13.0
skopeo copy docker://icr.io/cpopen/isf-operator-catalog:${FUSION_VERSION}-linux.amd64 dir:./images
```

This command downloads the image as a set of files and places them in the **images** directory (or another directory that you choose).
Note: There is a manifest file that is named **images/manifest.json**, and a signature file named **images/signature-1**. You reference both these files in the next step (in the command to verify the signature).
Log in to the **icr.io** entitlement registry to pull the image. Alternatively, you can pass your credentials directly to **skopeo** with **--src-creds iamapikey:<YOUR ENTITLEMENT KEY>**.

5. Verify the signature:

```
FUSION_VERSION=<version of Fusion being installed/verified>
sudo skopeo standalone-verify ./images/manifest.json icr.io/cpopen/isf-operator-catalog:${FUSION_VERSION}-linux.amd64
${fingerprint} ./images/signature-1
```

For example, to check the signatures on 2.13.0:

```
# FUSION_VERSION=2.13.0
sudo skopeo standalone-verify ./images/manifest.json icr.io/cpopen/isf-operator-catalog:${FUSION_VERSION}-linux.amd64
${fingerprint} ./images/signature-1
```

What's new

Each release offers new functions and improvements. IBM is constantly updating the infrastructure, security, and stability of IBM Fusion to improve your experience. Review this information for a high-level summary of the new features and changes in each release.

AI enhancements

Content-Aware Storage (CAS) service

v1.1.6

Improved network security and observability
: CAS enhances its security posture by providing deep observability for network policies, featuring real-time IBM Fusion UI event notifications, precise status conditions, and structured logging.

Use Content-Aware Storage (CAS) with Data Foundation clients
: Using CAS without a remotely mounted IBM Storage Scale filesystem is now supported on OpenShift® Hosted Control Plane hosted clusters running with Data Foundation StorageClients pointing to a centralized Provider Mode StorageCluster. For more information, see README.

v.1.1.5

Simplified CAS deployment with native RBD support
: In CAS v1.1.5, CAS backend uses the native RADOS Block Device (RBD) support introduced in IBM Storage Scale CNSA v6.0.1 to provision local cache filesystems by using RBD disks from Data Foundation. Enhanced automated deployment script eliminates additional setup steps and provides a more streamlined deployment experience. For details on setting up local cache filesystem, see Installing CAS without a remotely mounted IBM Storage Scale filesystem.

Network File System (NFS) data source support with CAS
: CAS introduces an NFS data source type, enabling you to bring enterprise data stored on NFS exports into your RAG applications. As with all other supported data source types, CAS uses IBM Docling Multimodal services and NVIDIA NeMo Retriever Extraction microservices for optimized content processing. By using IBM Storage Scale as a cache layer, CAS eliminates the need to maintain duplicate copies of data. This reduces storage costs and operational complexity while accelerating access to enterprise content for AI workloads. For more information, see Creating a data source.

Domain-level file access control
: CAS now supports domain-level configuration for file permission checking, allowing users to enable or disable file-level access control on a per-domain basis. This enhancement removes the need for separate scripts by providing built-in configuration through the user interface. With this capability, administrators can apply file-level controls selectively across domains while continuing to use an integrated identity provider with IBM Storage Scale for enforcing access policies. For more information, see Modifying file-level access control for a domain.

Zero-trust network policy enforcement
: Content-Aware Storage (CAS) now automatically enforces a zero-trust network communication model across its internal services by deploying and managing Kubernetes NetworkPolicies. This behavior is configurable through the CasInstall custom resource, allowing administrators to control policy enforcement as part of the deployment configuration. For more information, see Configuring network policies.

Installation and image mirroring precheck utilities

Improved image mirroring efficiency
: IBM Fusion now simplifies the image mirroring process by removing the dependency on the **ibm-pak** tool. You can now complete mirroring by using only the **oc-mirror** utility. In addition, the mirroring process is more efficient, as it copies only the required release images, resulting in reduced image size and lower storage requirements.

Security enhancements

IBM Fusion now supports Federal Information Processing Standards (FIPS)-certified cryptographic modules to ensure secure operations in regulated environments. This feature is optional and must be enabled before installing the OpenShift cluster. For more information, see IBM Fusion FIPS capabilities (Optional).

Simplified upgrade path for IBM Fusion

The upgrade process for IBM Fusion is streamlined. Users currently on 2.11.x can now directly upgrade to 2.13.0 on specific configurations, bypassing the need for intermediate 2.12.x releases. For details on supported configurations, see Enhanced upgrade experience.

Service management

Automatic storage class selection

IBM Fusion now automatically configures storage classes for its data services based on best-practice settings, reducing the need for manual setup and simplifying deployment. Manual configuration is required only when IBM Fusion storage is not used, providing flexibility for custom storage environments.

Serviceability enhancements

Simplified Call Home support

A new, simplified proxy configuration experience is available during setup, with support for multiple connectivity options:

- Direct connection
- Open proxy
- Proxy with basic authentication
- Proxy with certificate-based authentication

This ensures seamless connectivity in both restricted and unrestricted network environments from day one.

Enhanced log collection and automatic sanitization

IBM Fusion now supports targeted log collection for individual nodes and switches, reducing collection time and generating smaller, more relevant log packages for troubleshooting.

In addition, IBM Fusion includes automatic log package sanitization, which helps protect sensitive data when you collect diagnostic logs for troubleshooting and support. The system automatically omits or masks sensitive information from log packages, for example:

- MAC addresses
- IP addresses
- Subdomains
- Domains
- Gateways
- DNS and DHCP information

Automatic log sanitization is enabled by default, ensuring that diagnostic logs are safe to share with IBM Support without requiring manual cleanup. This capability reduces security risk while simplifying support interactions. For more information, see Automatic cleanup of log packages.

Reduced event noise and improved maintenance visibility

IBM Fusion now reduces event noise by automatically resolving compute, installation, and network-related critical issues once they are fixed, while enhanced deduplication minimizes duplicate alerts. During planned maintenance operations such as OpenShift upgrades, node maintenance initiated from the IBM Fusion UI, or switch upgrades, the system suppresses unnecessary critical alerts and degraded health indicators. This results in fewer, more relevant events and improved operational clarity.

IBM Lakehouse-Table service [Technology preview]

The Lakehouse-Table service is a Kubernetes-native solution that simplifies the creation and management of lakehouse infrastructure on IBM Fusion. This service provides a streamlined experience for administrators to provision and manage Data Lakehouse capabilities, including object storage integration and Iceberg catalog management, and integrates seamlessly with watsonx.data, enabling a comprehensive data lakehouse experience. For more information, see Lakehouse-Table [Technology preview].

Backup & Restore enhancements

Enhanced error messaging

Error messages now clearly indicate the specific component that triggered the error, enabling you to quickly identify and troubleshoot issues. This improvement reduces resolution time and allows you to potentially resolve problems independently or provide support with precise information to expedite the resolution process.

Optimized Backup & Restore pod resources

The Pod Resource Recommendation Tool is now available, enabling you to optimize the resources that are allocated to Backup & Restore pods. This tool analyzes historical data from a specified reference period to recommend optimal resource levels, ensuring that cluster resources are used efficiently. For more information, see Pod Resource Recommendation and Patch Tool.

Fusion Data Foundation support

IBM Fusion now includes support for Fusion Data Foundation 4.21. Key highlights in Fusion Data Foundation 4.21 include:

- Container Native Storage Access (CNSA) integration
- IBM Fusion Access for SAN integration
- Hybrid cloud support for disaster recovery
- Multi-volume consistency groups for RBD and CephFS volumes
- CephFS volumes support using non-default StorageClasses
- Enhanced protected applications list view
- Automatic resource cleanup for discovered VMs during DR operations
- Minimal IAM support for Multicloud Object Gateway
- Enhanced Multicloud Object Gateway database backup
- Developer access to object browser
- Internal mode RGW support with object browser
- Improved Multicloud Object Gateway performance using secondary database instance

- HCP clusters support enhancements
- HCP clusters deployment support for AWS and IBM Z
- Multus support for existing clusters
- IPv6 support for Multus networks
- IPv6 support for external mode
- Performance Plus enabled for Azure deployments
- Additional NooBaa bucket and replication metrics
- Clearer NooBaa capacity alerting
- New MDS xattr-latency alert
- Enhanced vCPU requirements for IBM Z and LinuxONE

For more information, see New features and Enhancements topics in *Fusion Data Foundation 4.21 release notes*. For known issues and limitations, see Known issues in 4.21 topic.

IBM Data Cataloging enhancements

Enhanced data classification

AI-powered content-based tagging
: IBM Data Cataloging introduces AI-powered content-based tagging to automatically classify unstructured data based on actual file content, rather than relying solely on metadata. By using large language models (LLMs) and external GPU acceleration, the solution analyzes data at scale and suggests relevant business tags with minimal manual effort. This capability standardizes content organization across multiple data sources, improving data discovery and governance. For more information, see Content-based AI-enabled autotagging.

Improved data governance
: IBM Data Cataloging enhances Content-Aware Storage (CAS) by propagating tags that are assigned to cataloged data. Tags, whether created manually, through auto-tagging policies, or through AI, are consolidated and propagated to the underlying data in CAS.

  This enrichment enables CAS to enrich content ingestion with tag attributes, resulting in more accurate semantic search and improved governance across domains. For more information, see Content-Aware Storage (CAS) APIs.

Simplified integration with Content-Aware Storage

One-click installation and enablement
: A new capability simplifies IBM Data Cataloging and Content-Aware Storage integration. This capability simplifies the integration between IBM Data Cataloging and Content-Aware Storage with a one-click installation and enablement experience. Users can deploy and activate IBM Data Cataloging directly from the CAS interface in just two steps: Install and Enable.

  This streamlined setup enhances operational efficiency and allows CAS to use advanced scanning and filtering capabilities for more granular data ingestion control. For more information, see One-click installation and enablement integration experience.

Data insights dashboards
: IBM Data Cataloging and CAS now include data insights dashboards that provide immediate visibility into data once sources are connected and scanned. This experience is enabled when both IBM Data Cataloging and CAS are integrated. The provided insights surface key indicators of data composition on each established data source. For more information, see IBM Data Cataloging and Content-Aware Storage insights.

Security fixes and updates
: Enhance IBM Data Cataloging service stability and protect against known vulnerabilities.

New currency support

- Content-Aware Storage 1.1.6
- OpenShift Container Platform 4.21
- IBM Storage Scale Data Management Edition or Global Data Platform service 6.0.1.0 and 6.0.0.2
- Backup & Restore 2.13.0
- IBM Data Cataloging 2.5.3
- For supported IBM Cloud Paks versions, see IBM Cloud Paks support for IBM Fusion.
- For supported IBM Maximo® versions, see IBM Maximo® Applications support for IBM Fusion.

For a complete list of currency support, see Support matrix for IBM Fusion versions.

Deploying IBM Fusion

Prerequisites and procedure to install IBM Fusion on OpenShift® Container Platform that runs on On-premises VMware, On-premises Bare Metal, Linux on IBM Z, Microsoft Azure, Amazon Web Services, IBM Cloud®, Google Cloud, and IBM Power Systems.

- **Prerequisites**
  Prerequisite that you need for a successful setup and installation of IBM Fusion.
- **Installing IBM Fusion on On-premises VMware**
  Procedure to install IBM Fusion on OpenShift Container Platform that runs on On-premises VMware.
- **Installing IBM Fusion on On-premises Bare Metal**
  Procedure to install IBM Fusion on OpenShift Container Platform that runs on On-premises Bare Metal.
- **Installing IBM Fusion on On-premises Linux on IBM Z and zCX**
  Procedure to install IBM Fusion on OpenShift Container Platform that runs on On-premises Linux on IBM Z and zCX.
- **Installing IBM Fusion on IBM Cloud**
  Procedure to install IBM Fusion on OpenShift Container Platform that runs on IBM Cloud.
- **Installing IBM Fusion on Amazon Web Services**
  Procedure to install IBM Fusion on OpenShift Container Platform that runs on Amazon Web Services and Amazon Web Services ROSA.

- **Installing IBM Fusion on Microsoft Azure**
  Procedure to install IBM Fusion on OpenShift Container Platform that runs on Microsoft Azure.
- **Installing IBM Fusion on On-premises IBM Power Systems**
  Procedure to install IBM Fusion on OpenShift Container Platform that runs on On-premises IBM Power Systems.
- **Installing IBM Fusion on Google Cloud**
  Procedure to install IBM Fusion on OpenShift Container Platform that runs on Google Cloud.
- **Installing IBM Fusion from enterprise registry**
  Prerequisites and procedure to install IBM Fusion on OpenShift Container Platform.
- **Deploying IBM Fusion using CLI**
  Use the instructions to deploy the IBM Fusion using the command-line interface (CLI).
- **Validating IBM Fusion installation**
  Steps to confirm the success of the installation.
- **Uninstalling IBM Fusion**
  Steps to uninstall IBM Fusion from Red Hat® OpenShift web console.
- **Installation and upgrade issues**
  Use the troubleshooting tips and tricks in IBM Fusion installation and upgrade.

Prerequisites

Prerequisite that you need for a successful setup and installation of IBM Fusion.

Procedure

1. Go through the system requirements and confirm whether you meet all the requirements. For detailed information about system requirements, see System requirements.
2. For the prerequisites of individual IBM Fusion services, go through their respective *Before you begin* section.
   For information about the services, see Fusion services.
3. Ensure that you have the infrastructure and supported version of Red Hat OpenShift Container Platform.
   Note:
   - You can use Red Hat OpenShift Kubernetes Engine (OKE) as an alternative for OpenShift Container Platform. The supported version of OKE is same as OpenShift Container Platform.
   - IBM Fusion provides support for Red Hat OpenShift Virtualization Engine (OVE) as an alternative for OpenShift Container Platform. If you do not need all the capabilities of OpenShift for initial migrations of VMs and want a simplified and less costly alternative edition of self-managed OpenShift, then you can use OVE for running VMs. Consider the following inclusions and restrictions in the entitlement before you proceed with OVE:
     - OVE restricts your application instances to run as a VM only. RHEL entitlements for the VMs must be purchased separately if needed.
     - End user container workloads in OVE can only run on independent clusters deployed on the virtual machines which must be provided with separate core-based subscriptions for OpenShift.
     - Infrastructure workloads such as storage drivers, backup applications, forwarding agents, Red Hat Advanced Cluster Management for Kubernetes, and Ansible Automation Platform, are permitted to run as containers in OVE.
     - Containers that are qualified to run on infrastructure nodes according to the self-managed Red Hat OpenShift subscription guide can also be run on infrastructure nodes in an OVE cluster.
     - Fusion Data Foundation including MCG, Scale CNSA, Backup & Restore Spoke, Fusion Control Tower are permitted to run on infrastructure nodes.
     - IBM Data Cataloging, Backup & Restore Hub and Fusion Base can be deployed as containers on OVE but must run on normal worker nodes.

   For the supported versions of OpenShift Container Platform, see Support matrix.

   Note: If you want to install Data Foundation storage in consumer mode, then OpenShift Container Platform must also be the same version.
4. Obtain IBM Entitlement Registry Key.
   For the steps to obtain the key, see Obtaining entitlement key.
5. Create a pull secret.
   For the actual steps to generate, see Creating image pull secret.
   For actual steps to create a pull secret for installing IBM Fusion on IBM cloud, see Creating image pull secret for IBM Cloud based installation.
6. If you plan to deploy IBM Fusion on a OpenShift Container Platform cluster with a cluster wide proxy configured, then ensure you add these hosts to your allowlist:
   - registry.redhat.io
   - quay.io
   - cp.icr.io
   - icr.io
   - redhat.com
   - registry.connect.redhat.com
   - catalog.redhat.com
   - access.redhat.com
   - cloud.redhat.com
   - cdn02.quay.io
   - registry.access.redhat.com
   - dd0.icr.io
   - dd2.icr.io
   - dd4.icr.io
   - dd6.icr.io

   Note: This step is applicable only for on-premise deployments.
7. Label GPU nodes to avoid IBM Fusion pods from being scheduled on them. It frees up resources on the GPU nodes to be used by AI workloads.
   For detailed guidance on implementation, see AI workloads on IBM Fusion.

- **Obtaining entitlement key**
  Entitlement keys determine whether IBM Fusion software operator can automatically pull the required container native images. During installation, image pull failures may occur due to an invalid entitlement key or a key belonging to an account that does not have entitlement to IBM Fusion.
- **Creating image pull secret**
  Create an image pull secret to enable the OpenShift cluster to authenticate with the IBM Entitled Registry. The image pull secret provides the credentials for pulling Docker images from the IBM Entitled Registry.

- **Creating image pull secret for IBM Cloud based installation**
  With OpenShift Container Platform, you can create a global image pull secret that each worker node in the cluster can use to pull images from a private registry and you can set up your Red Hat OpenShift on IBM Cloud cluster to pull entitled software.
- **IBM Fusion FIPS capabilities (Optional)**
  IBM Fusion uses Federal Information Processing Standards (FIPS)-certified cryptographic modules to ensure secure operations in regulated environments. This is provided by a combination of OpenShift configuration and a key feature of the IBM Fusion application.
- **System requirements**
  The system requirements for installing IBM Fusion software.
- **SecurityContextConstraints**
  Red Hat OpenShift `SecurityContextConstraints` requirement controls pod permissions. These permissions include pod actions and resources that it can access.

Obtaining entitlement key

Entitlement keys determine whether IBM Fusion software operator can automatically pull the required container native images. During installation, image pull failures may occur due to an invalid entitlement key or a key belonging to an account that does not have entitlement to IBM Fusion.

Procedure

1. Log in to the IBM container software library with an IBM id and a password that is associated with the entitled software.
2. In the navigation bar, click Get entitlement key.
3. On the Access your container software page, click Copy key to copy the generated entitlement key.
4. Save the key to a secure location for future use.

Creating image pull secret

Create an image pull secret to enable the OpenShift® cluster to authenticate with the IBM Entitled Registry. The image pull secret provides the credentials for pulling Docker images from the IBM Entitled Registry.

Before you begin

- As a prerequisite to run base64, you must install base64 or `jq`.
  Download the `jq` JSON processor command line package. To download the `jq` JSON processor package, see Download the `jq` JSON processor command line package. You use `jq` to combine the JSON value of the default global pull secret with the private registry pull secret that you want to add.
- Ensure that you have access to https://cloud.ibm.com/docs/openshift?topic=openshift-access_cluster.

Procedure

1. Run the following command to log in to OpenShift system:

   ```
   oc login --token=<API token> --server=https://<server name>:6443
   ```

2. Create a base64 encoded string of the credentials used to access the image registry.
   Linux example:

   ```
   echo -n "cp:GENERATED_ENTITLEMENT_KEY" | base64 -w0
   ```

   Windows example:

   ```
   echo -n "cp:GENERATED_ENTITLEMENT_KEY" | base64 -e
   ```

   Note: Replace `GENERATED_ENTITLEMENT_KEY` with the entitlement key that you generated in Obtaining entitlement key.
3. Create an `authority.json` to include the base64 encoded string of your credentials (created in the previous step), user name (to access `cp.icr.io` repository), and generated entitlement key for the IBM Cloud Container Registry.
   Windows or Linux example:

   ```
   {
     "auth": "BASE64_ENCODED_ENTITLEMENT_KEY"
   }
   ```

   Note: Replace `BASE64_ENCODED_ENTITLEMENT_KEY` with the value of base64 encoded entitlement key got from the previous step.
4. The following step takes the `authority.json` and includes it as a new authority in your `.dockerconfigjson`, stored as a `temp_config.json`.

   ```
   oc get secret/pull-secret -n openshift-config -ojson | jq -r '.data[".dockerconfigjson"]' | base64 -d  | jq '.
   []."cp.icr.io" += input' - authority.json > temp_config.json
   ```

   Note: The referenced `jq` command is not installed on OpenShift Container Platform clusters by default. Run the following command to install jq:

   ```
   yum install jq
   ```

   The `yum install jq` command is for Red Hat® Enterprise Linux systems. For other operating system, use the appropriate commands to install `jq`.
5. Use the contents of the `temp_config.json` file and apply the updated config to the OpenShift cluster.

   ```
   oc set data secret/pull-secret -n openshift-config --from-file=.dockerconfigjson=temp_config.json
   ```

6. To verify whether your pull-secret is updated with your new authority, run the following command and confirm that your authority is present.

```
oc get secret/pull-secret -n openshift-config -ojson | jq -r '.data[".dockerconfigjson"]' | base64 -d
```

The updated config is now rolled out to all the nodes in the OpenShift cluster.

7. When the global pull secret is updated, enter the following command to remove the temporary files that were created.

```
rm authority.json temp_config.json
```

Creating image pull secret for IBM Cloud® based installation

With OpenShift® Container Platform, you can create a global image pull secret that each worker node in the cluster can use to pull images from a private registry and you can set up your Red Hat® OpenShift on IBM Cloud cluster to pull entitled software.

Before you begin

- Download the **jq** JSON processor command line package. To download the **jq** JSON processor package, see Download the **jq** JSON processor command line package. You use **jq** to combine the JSON value of the default global pull secret with the private registry pull secret you want to add.
- Ensure that you have access to https://cloud.ibm.com/docs/openshift?topic=openshift-access_cluster.

About this task

To add private registries, edit the global **pull-secret** in the **openshift-config** project. You can set up your Red Hat OpenShift on IBM Cloud cluster to pull entitled software, which is a collection of protected container images that are packaged in Helm charts that you are licensed to use by IBM.

Important: Do not replace the global pull secret with a pull secret that does not have credentials to the default Red Hat registries. If you do, the default Red Hat OpenShift components that are installed in your cluster, such as the OperatorHub, might fail because they can't pull images from these registries.
You must only update the global pull secret to add more auths and never overwrite.

Procedure

1. To add private registries, edit the global **pull-secret** in the **openshift-config** project as follows:
   a. Create a secret value that holds the credentials to access your private registry and store the decoded secret value in a JSON file. When you create the secret value, the credentials are automatically encoded to base64. By using the **--dry-run** option, the secret value is created only and no secret object is created in your cluster. The decoded secret value is then stored in a JSON file to later use in your global pull secret.

   ```
   oc create secret docker-registry <secret_name> --docker-server=<registry_URL> --docker-username=<docker_username> --docker-password=<docker_password> --docker-email=<docker_email> --dry-run=client --output="jsonpath={.data.\.dockerconfigjson}" | base64 --decode > myregistryconfigjson
   ```

   The following parameters are mandatory.

   - --namespace <project>
     The Red Hat OpenShift project of your cluster where you want to use the secret and deploy containers to. To list available projects in your cluster, run the command.

     ```
     oc get ns
     ```

   - <secret_name>
     The name that you want to use for your image pull secret.
   - --docker-server <registry_URL>
     The URL to the registry where your private images are stored.
   - --docker-username <docker_username>
     The username to log in to your private registry.
   - --docker-password <token_value>
     The password to log in to your private registry, such as a token value.
   - --docker-email <docker-email>
     If you have Docker account, enter your docker email address. If you don't have one, enter a fictional email address, such as **a@b.c**. This email is required to create a kubernetes secret, but is not used after creation.

   The following parameters are not mandatory.

   - --dry-run=true
     Include this flag to create the secret value only, and not create and store the secret object in your cluster.
   - --output="jsonpath={.data.\.dockerconfigjson}"
     Get only the **.dockerconfigjson** value from the data section of the Kubernetes secret.
   - | base64 --decode > myregistryconfigjson
     Download the decoded secret data to a local **myregistryconfigjson** file.

   b. Retrieve the decoded secret value of the default global pull secret and store the value in a **dockerconfigjson** file.

   ```
   oc get secret pull-secret -n openshift-config
    --output="jsonpath={.data.\.dockerconfigjson}" | base64
   --decode > dockerconfigjson
   ```

   c. Combine the downloaded private registry pull secret **myregistryconfigjson** file with the default global pull secret **dockerconfigjson** file.

```
jq -s '.[0] * .[1]' dockerconfigjson myregistryconfigjson >
dockerconfigjson-merged
```

d. Update the global pull secret with the combined `dockerconfigjson-merged` file.

```
oc set data secret/pull-secret -n openshift-config --from
-file=.dockerconfigjson=dockerconfigjson-merged
```

e. Verify that the global pull secret is updated. Check that your private registry and each of the default Red Hat registries are in the output of the following command.

```
oc get secret pull-secret -n openshift-config
--output="jsonpath={.data.\.dockerconfigjson}" | base64
--decode
```

Sample output

```
{
    "auths": {
        "cloud.openshift.com": {
            "auth": "<encoded_string>",
            "email": "email@example.com"
        },
        "quay.io": {
            "auth": "<encoded_string>",
            "email": "email@example.com"
        },
        "registry.connect.redhat.com": {
            "auth": "<encoded_string>",
            "email": "email@example.com"
        },
        "registry.redhat.io": {
            "auth": "<encoded_string>",
            "email": "email@example.com"
        },
        "<private_registry>": {
            "username": "iamapikey",
            "password": "<encoded_string>",
            "email": "email@example.com",
            "auth": "<encoded_string>"
        }
    }
}
```

f. To pick up the global configuration changes, reload all the worker nodes in your cluster.

i. Note the ID of the worker nodes in your cluster.

```
ibmcloud oc worker ls -c <cluster_name_or_ID>
```

ii. Reload each worker node. You can reload multiple worker nodes by including multiple `-w` flags, but make sure to keep enough worker nodes running at the same time for your apps to avoid an outage.
Note: For IBM VPC type clusters, the Reload option is not available, instead you can use the Replace option.

```
ibmcloud oc worker reload -c <cluster_name_or_ID>
-w <workerID_1> -w <workerID_2>
```

g. After the worker nodes are back in a healthy state, verify that the global pull secret is updated on a worker node.

i. Start a debugging pod to log in to a worker node. Use the Private IP that you retrieved earlier for the `<node_name>`.

```
oc debug node/<node_name>
```

ii. Change the root directory to the host so that you can view files on the worker node.

```
chroot /host
```

iii. Verify that the Docker configuration file has the registry credentials that match the global pull secret that you set.

```
vi /.docker/config.json
```

2. Setting up a cluster to pull entitled software as follows:
Entitled software is stored in a special IBM Cloud Container Registry `cp.icr.io` domain. To access this domain, you must create an image pull secret with an entitlement key for your cluster and add this image pull secret to the Kubernetes service account of each project where you want to deploy this entitled software.

a. Get the entitlement key for your entitled software library.

i. Log in to MyIBM.com and scroll to the Container software library section. Click View library.

ii. From the Access your container software > Entitlement keys page, click Copy key. This key authorizes access to all the entitled software in your container software library.

b. In the project that you want to deploy your entitled containers, create an image pull secret so that you can access the `cp.icr.io` entitled registry. Use the entitlement key that you previously retrieved as the `--docker-password` value. For more information, see Accessing images that are stored in other private registries.

```
oc create secret docker-registry entitled-cp-icr-io --docker-server=cp.icr.io --docker-username=cp --docker-password=
<entitlement_key> --docker-email=<docker_email> -n <project>
```

c. Add the image pull secret to the service account of the namespace so that any container in the project can use the entitlement key to pull entitled images. For more information, see Using the image pull secret to deploy containers.

```
oc patch -n <project> serviceaccount/default --type='json' -p='[{"op":"add","path":"/imagePullSecrets/-","value":
{"name":"entitled-cp-icr-io"}}]'
```

d. Create a pod in the project that builds a container from an image in the entitled registry.

```
oc run <pod_name> --image=cp.icr.io/<image_name> -n <project> --generator=run-pod/v1
```

e. Check that your container was able to successfully build from the entitled image by verifying that the pod is in a Running status.

```
oc get pod <pod_name> -n <project>
```

f. To pick up the global configuration changes, reload all the worker nodes in your cluster.

i. Note the ID of the worker nodes in your cluster.

```
ibmcloud oc worker ls -c <cluster_name_or_ID>
```

ii. Reload each worker node. You can reload multiple worker nodes by including multiple `-w` flags, but make sure to keep enough worker nodes running at the same time for your apps to avoid an outage.
Note: For IBM VPC type clusters, the Reload option is not available, instead you can use the Replace option.

```
ibmcloud oc worker reload -c <cluster_name_or_ID>
-w <workerID_1> -w <workerID_2>
```

g. After the worker nodes are back in a healthy state, verify that the global pull secret is updated on a worker node.

i. Start a debugging pod to log in to a worker node. Use the Private IP that you retrieved earlier for the `<node_name>`.

```
oc debug node/<node_name>
```

ii. Change the root directory to the host so that you can view files on the worker node.

```
chroot /host
```

iii. Verify that the Docker configuration file has the registry credentials that match the global pull secret that you set.

```
vi /.docker/config.json
```

IBM Fusion FIPS capabilities (Optional)

IBM Fusion uses Federal Information Processing Standards (FIPS)-certified cryptographic modules to ensure secure operations in regulated environments. This is provided by a combination of OpenShift® configuration and a key feature of the IBM Fusion application.

Key features

While IBM Fusion does not undergo standalone FIPS certification, it operates in strict FIPS compliance by offloading cryptographic operations to the FIPS-certified modules provided by the host environment.

When deployed on a correctly configured FIPS-enabled cluster, IBM Fusion provides:

- Data in transit: All network traffic is encrypted by using FIPS-compliant algorithms.
- Data at rest: Data is encrypted at rest by using FIPS-compliant methods, contingent on the customer's storage provider configuration.
- Secure key management: Cryptographic keys are generated, stored, and managed securely in accordance with FIPS standards.

Cluster provisioning

Follow these steps while you provision the OpenShift cluster:
Warning: FIPS mode must be enabled during the initial installation of the OpenShift cluster. Upgrading or reconfiguring an existing, non-FIPS cluster to FIPS mode is strictly unsupported.

1. Enable FIPS mode.
   Follow the Red Hat® documentation to enable FIPS at the cluster level.
2. Use FIPS-compliant local storage.
   You have multiple options that include, but are not limited to:
   - Self-encrypting drives
   - LUKS. For more information, see the Red Hat documentation.
3. Enable IPsec (`Full` mode): Configure IPsec to secure node-to-node network traffic.
   For more information, see the Red Hat documentation.
4. Encrypt etcd: Enable etcd encryption to protect the key-value store of the cluster.
   For more information, see the Red Hat documentation.

Limitation

The following services are excluded from the FIPS support:

- Content-Aware Storage (CAS) and Data Lakehouse do not support FIPS mode because both IBM Storage Scale Remote Mount and the IBM Storage Scale CNSA driver do not support FIPS mode.

System requirements

The system requirements for installing IBM Fusion software.

The IBM Fusion operators and your workloads run on OpenShift® Container Platform that in turn uses 64-bit Intel, AMD x86, Linux on IBM Z, or Power hardware architecture. For more information about Fusion data services on the different deployment platforms of IBM Fusion, see IBM Fusion services and platform support matrix.

The following table lists the system requirements for IBM Fusion software:
Note: Though there is no minimum number of nodes requirement, the following table provides data based on three control nodes and three compute nodes cluster. However, sizing changes based on the number of nodes for components with "Per node".
Fusion Data Foundation can be deployed on Infra or worker nodes, whereas all other IBM Fusion services including Base must be deployed on worker nodes.

Component	vCPUs	Memory	Storage
IBM Fusion Base Note: For CAS, you need a GPU-based solution.	Request CPU: 4 Limit CPU: 10	Request memory: 4 GiB Limit memory: 13 GiB	Overall storage is a minimum of 500 MB.
Fusion Data Foundation			
Fusion Data Foundation storage	Internal mode (Local and dynamic) Request CPU: 22+2* (total count of Object Storage Daemon) Limit CPU: 26+2* (total count of Object Storage Daemon) External mode Request CPU: 6 Limit CPU: 6 **For IBM Power Systems**: Internal mode 48 CPU (logical) External mode 24 CPU (logical)	Internal mode (Local and dynamic) Request memory: 84+5* (total count of Object Storage Daemon) GiB Limit memory: 88+5* (total count of Object Storage Daemon) GiB External mode Request memory:16 GiB Limit memory: 16 GiB **For IBM Power Systems:** Internal mode 192 GiB memory External mode 48 GiB memory	Minimum three storage Nodes For more information about Fusion Data Foundation, see Important considerations when deploying Red Hat OpenShift Data Foundation. For IBM Power Systems: Internal 3 storage devices, each with additional 500 GB of disk
Fusion Data Foundation MCG	Core CPU: 1 Database CPU: 1 Endpoint CPU: 1	Core Memory: 4 GiB Database Memory: 4 GiB Endpoint Memory: 2 GiB	NA
Global Data Platform	Per node Request CPU: 10% (total vCPU of the node) Cluster Request CPU: 3 Limit CPU: 14	Per node Request memory: 10% (total memory of the node) Cluster Request memory: 14 Limit memory: 35	For more information about IBM Spectrum Scale Container Native Storage Access, see Hardware requirements for IBM Storage Scale Container Native Storage Access.
Backup & Restore			
Backup & Restore Base Hub install requirements (assuming 3 node cluster)	• Request CPU: 4.5 • Limit CPU: 23	• Request Memory: 9.5 GiB • Limit Memory: 33 GiB	• Request: Ephemeral Storage: 3.7 GiB PVC Storage: 200 GiB • Limit: Ephemeral Storage: 8 GiB PVC Storage: NA

Component	vCPUs	Memory	Storage
Backup & Restore Base Spoke install requirements (3 node cluster). For more information scaling the hub, see Backup & restore hub performance and scaling.	• Request CPU: 2.3 • Limit CPU: 9.15	• Request Memory: 2 GiB • Limit Memory: 11 GiB	• Request: Ephemeral Storage: 850 MiB PVC Storage: 10 GiB • Limit: Ephemeral Storage: 1900 MiB PVC Storage: NA
Each additional node for Backup & Restore (Beyond the first three) Applicable for both Hub and Spoke)	• Request CPU: 0.2 • Limit CPU: 0.4	• Request Memory: 150 MiB • Limit Memory: 800 MiB	• Request Ephemeral Storage: 50 MiB • Limit Ephemeral Storage: 250 MiB
Per Job Additional Requirements: Data Mover (1 per PVC, max 5 per node)	Request CPU: 2 Limit CPU: 4	Request Memory: 2 GiB Limit Memory: 10 GiB	Request Ephemeral Storage: 5 GiB Limit Ephemeral Storage: 5 GiB
Per Job Additional Requirements: Legacy Restic Data Mover (1 per PVC, max 3 per node)	Request CPU: 0.1 Limit CPU: 2	Request Memory: 4,000 MiB Limit Memory: 15,000 MiB	Request Ephemeral Storage: 400 MiB Limit Ephemeral Storage: 2,000 MiB
Per Job Additional Requirements: Maintenance (after every job and once per day per application/policy combination)	Request CPU: 0.5 Limit CPU: 4	Request Memory: 500 MiB Limit Memory: 4 GiB	Request Ephemeral Storage: 5 GiB Limit Ephemeral Storage: 5 GiB
IBM Data Cataloging	Request CPU: 21 Limit CPU: 78	Request memory: 27 GiB Limit memory: 126 GiB	120 GB Minimum For more information about IBM Data Cataloging, see Installing IBM Data Cataloging.
Content-Aware Storage (CAS)			
Content-Aware Storage with less than 12 TB of ingested data Note: For Content-Aware Storage with more than 12 TB of Ingested Data, contact IBM Sales representatives for your intended workloads.	Request: 9 Limit: 25	Request: 22 GiB Limit: 24 GiB	The minimum required filesystem size is 550 GB. For more information about IBM Spectrum Scale Container Native Storage Access, see Hardware requirements for IBM Storage Scale Container Native Storage Access.
`nv-ingest` Requirements	Request: 30 Limit: 60	Request: 90 GiB Limit: 290 GiB	100 GB
`docling_multimodal` - Additional requirements	Request: 4.5 Limit: 21	Request: 13 GiB Limit: 44 GiB	100 GB
Minimum requirements for each additional Collection	Request: 1 Limit: 2	Request: 10 GiB Limit: 10 GiB	1 GB

GPU resources

Configuration	GPU worker nodes	Non-GPU worker nodes	Production with Multi-Instance GPU (MIG)* enabled	Production without Multi-Instance GPU (MIG) enabled
Content-Aware Storage with less than 12 TB of ingested data	1	2	2 or more MIG Capable NVIDIA GPUs (A100 80GB, H100, H200, RTX PRO 6000)	1 NVIDIA GPU for each NVIDIA NIM service** configured (L40S, A100 H100, H200, RTX PRO 6000)

* For more information about MIG, see Configuring a Multi-Instance GPU (MIG) with CAS.

** For more information about the NVIDIA NIM services, see Select NVIDIA Inference Microservices (NIMs).

The following table lists the component versions:

Component	Versions
On-premises VMware	Version 7.0
On-premises IBM z/OS Container Extensions (IBM zCX)	All the zCX hypervisor versions on which OpenShift Container Platform is supported.
Red Hat® OpenShift Container Platform	4.18, 4.19, 4.20, and 4.21. Note: The console plugin does not support the OpenShift Container Platform versions of Fusion Backup & Restore service. For more information, see Guidance for Fusion Native Console plugin OCP version.
Red Hat OpenShift Kubernetes Engine (OKE)	You can use OKE as an alternative for OpenShift Container Platform. The supported version of OKE is same as OpenShift Container Platform.
Red Hat OpenShift Virtualization Engine (OVE)	IBM Fusion provides support for OVE as an alternative for OpenShift Container Platform. The supported version of OVE is same as OpenShift Container Platform. For more information about the limitations and considerations of using OVE, see OVE considerations.

Component	Versions
Storage	Fusion Data Foundation 4.18, 4.19, 4.20, and 4.21 Note: For deployment platforms and supported services, see Support matrix for IBM Fusion versions. IBM Storage Scale Note: IBM Fusion supports IBM Storage Scale 5.2.3.x. Remote mount of IBM Storage Scale storage: • IBM Storage Scale remote storage cluster release 5.1.9.0 or higher. • The IBM Fusion cluster accesses the storage owned by an IBM Storage Scale storage cluster by using a remote mount. The IBM Storage Scale file system version of the owning storage cluster cannot be newer than version 37.00. To determine the version of your IBM Storage Scale cluster, run the **mmdiag --version** command on the IBM Storage Scale cluster. To determine the version of your IBM Storage Scale file system, run the **mmlsfs all -V** command. For further information, including installation/upgrade instructions for the remote IBM Storage Scale cluster, see IBM Storage Scale documentation.
Backup & Restore	IBM Fusion supports backup and recovery operations on Red Hat OpenShift Container Platform environment with any storage provider that implemented the Container Storage Interface (CSI) driver and meets the following prerequisites: • The Container Storage Interface driver must be v1 or higher (no support for *alpha* or *beta* versions) • The Container Storage Interface driver must support Volume Snapshot and Restore. • StorageClass must be created with `volumeBindingMode: Immediate` or `volumeBindingMode: WaitForFirstConsumer` and `allowVolumeExpansion: true`. • PersistentVolume (PV) must be created with `volumeMode: Filesystem` or `volumeMode: Block` Note: It is also recommended that the `VolumeSnapshotClass` deletion policy is configured to delete snapshots after the corresponding `VolumeSnapshotContent` is deleted (`deletionPolicy: Delete`). Note: For backups of PVCs provisioned on IBM Storage Scale, snapshots are be created only from independent fileset-based persistent volume claims (PVCs). PVCs that are based on lightweight directories and dependent file sets are not supported.

SecurityContextConstraints

Red Hat® OpenShift® `SecurityContextConstraints` requirement controls pod permissions. These permissions include pod actions and resources that it can access.

You do not have to set any of these constraints as all non-default constraints are automatically set by the product installer. To view the set of `SecurityConstantConstraints`, log in to OpenShift Container Platform and run the following command:

```
oc get scc
```

For more information about `SecurityContextConstraints`, see https://docs.redhat.com/en/documentation/openshift_container_platform/4.21/html/authentication_and_authorization/managing-pod-security-policies.

The IBM Storage Scale specific constraints are as follows:
Note: IBM Fusion 2.11.0 supports IBM Storage Scale 5.2.3.0.

```
NAME                                    PRIV    CAPS          SELINUX     RUNASUSER          FSGROUP     SUPGROUP    PRIORITY
READONLYROOTFS   VOLUMES
ibm-spectrum-scale-privileged           true    ["*"]         RunAsAny    RunAsAny           RunAsAny    RunAsAny    <no
value>   false           ["*"]
restricted-ibm-spectrum-protect-plus-ns   false   <no value>    MustRunAs   MustRunAsRange     MustRunAs   RunAsAny    <no
value>   false           ["configMap","downwardAPI","emptyDir","persistentVolumeClaim","projected","secret"]
spectrum-scale-csiaccess                true    []            RunAsAny    RunAsAny           MustRunAs   RunAsAny    <no
value>   false           ["configMap","downwardAPI","emptyDir","hostPath","persistentVolumeClaim","projected","secret"]
data-protection-scc-ibm-backup-restore    true    <no value>    RunAsAny    RunAsAny           RunAsAny    RunAsAny    <no
value>   false
data-protection-scc-ibm-dataprotection    true    <no value>    RunAsAny    RunAsAny           RunAsAny    RunAsAny    <no
value>   false
data-protection-scc-ibm-dp              true    <no value>    RunAsAny    RunAsAny           RunAsAny    RunAsAny    <no
value>   false
guardian-dm-datamover-scc               true    ["SYS_ADMIN"] RunAsAny    RunAsAny           RunAsAny    RunAsAny    <no
value>   false           ["configMap","emptyDir","hostPath","persistentVolumeClaim","secret"]
velero-privileged                       true    ["*"]         RunAsAny    RunAsAny           RunAsAny    RunAsAny    <no
value>   false           ["*"]
```

Enterprise registry for IBM Fusion installation

During the installation process, IBM Fusion installs IBM Fusion software using images hosted in the IBM entitled registries. If you want to use your enterprise registry, you can install IBM Fusion software from images that are maintained in a container registry that you manage.

In IBM Fusion, JFrog Artifactory is the supported enterprise registry.

Though the experience is same as using the IBM entitled and Red Hat® registry, you must do additional steps to set up images in your private enterprise registry.

You are responsible for setting up and populating this private enterprise registry.

These mirroring procedures guide you to download the images of IBM Fusion and all its services at once or individually depending on your specific needs. Before you initiate the mirroring process, verify that all prerequisites are met to ensure its success, regardless of whether you choose the end-to-end procedure or selective procedure.

Mirroring prerequisites

Complete the following perquisites before you mirror images of IBM Fusion and its services.

Install `oc` CLI and related mirroring tools

1. Install Docker or Podman
2. Install oc command tool from any of the Red Hat OpenShift Container Platform cluster:
   a. Log in to the system from where you want to run commands.
   b. Log in to OpenShift Container Platform.
   c. Click bell icon and select **Command line tools**.
   d. Click the appropriate **oc download** option based on your operating system.
   Alternatively, you can also download the platform based oc client from the following OpenShift Container Platform link:
   https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.18.x/.
3. Install the `oc-mirror` CLI plug-in. For more information about `oc-mirror` CLI plug-in installation, see Installing the oc-mirror OpenShift CLI plugin.
   Note: Ensure that you install the `oc-mirror` CLI plug-in of **4.20.6** version due to RH bug.

Configure common environment variables

Define the following environment variables for your target container registry. Each of the subsequent steps makes use of these values to mirror all the related components.

```
export LOCAL_ISF_REGISTRY="<Your container registry host>:<port>"
export LOCAL_ISF_REPOSITORY="<Your image path>"
export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
```

`LOCAL_ISF_REGISTRY` is your target container registry.

`LOCAL_ISF_REPOSITORY` is the image path in which you want to mirror the images. You can choose your own repository paths. For example, sds-images/isf or sds-images. See the following sample values:

```
export LOCAL_ISF_REGISTRY="registryhost.com:443"
export LOCAL_ISF_REPOSITORY="mirror-all-fusion"
export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
```

Considerations

- Repository and target path must be same for IBM Fusion and IBM Storage Scale.
- Run all commands as root user.
- In the commands, replace `<your enterprise registry>` with your enterprise registry and its corresponding pull-secret.
- If you use a non-default port (other than 443), then append a colon and port number after your enterprise registry value. For example, `<Your enterprise registry>:9443`.
- You must mirror images to a single repository, a single registry, and a single path.
- Depending on which IBM Fusion services you want to enable you can choose to only mirror images for that service, however you must mirror IBM Fusion images. Review the IBM Fusion and its services support matrix for platform compatibility. For more information, see IBM Fusion services and platform support matrix.
- The offline mirroring process includes applying `ImageDigestMirrorSet` or `ImageContentSourcePolicy` files to your cluster. Any changes to the IDMS or ICSP on clusters triggers a `MachineConfigPool` rollout, which can cause nodes to reboot. This includes adding, updating, or deleting IDMS or ICSP files.
  There are two options for offline upgrade mirroring:

  Mirroring images to the same location as the previous offline installation
  This gives the IDMS or ICSP files that are identical to your previous installation, and you can choose to not apply the IDMS or ICSP files or `oc apply` the file and OpenShift® recognizes there is no change in content and exit gracefully.

  Mirror images to a different location as the previous offline installation
  Because the generated IDMS or ICSP is tailored specifically to the target offline registry, mirroring to a new location every time gives a different IDMS or ICSP every time. Applying a new IDMS or ICSP triggers the MCP rollout and contains the potential to reboot nodes, which can be disruptive to the health of the OpenShift cluster.
  Additionally, if you have done multiple offline upgrades using this method, then your cluster likely have multiple ICSP or IDMS files, some of which are old and unused. Again, deleting a IDMS or ICSP can trigger a disruptive MCP rollout. It is recommended to not delete any IDMS or ICSP files from the cluster until you have completed your upgrade to your target version of IBM Fusion HCI.
  However, it is recommended you to edit the generated IDMS and change the `metadata.name` to unique name to avoid overwriting an existing IDMS on your cluster.

End-to-end mirroring of IBM Fusion and its services images to the enterprise registry

If you are planing a disconnected or offline installation, then you must mirror images to your enterprise registry. You can use this procedure to complete an integrated mirroring of IBM Fusion and its services **Backup & Restore**, **IBM Data Cataloging**, and **Content-Aware Storage**. Otherwise, you can mirror them individually per your requirement.

Before you begin

- Before you mirror, go through the prerequisites. See Mirroring prerequisites.

- Use either podman or docker commands to login to the registries. The examples in the procedure used docker; if you are using podman, substitute `podman` for `docker` in the commands.
- Configure Common Environment Variables:
  Important: Ensure you review the offline upgrade considerations before proceeding with upgrade mirroring.
  Define the following environment variables for your target container registry. These values will be used in each of the subsequent steps to mirror all the related components.

  ```
  export LOCAL_ISF_REGISTRY="<Your container registry host>:<port>"
  export LOCAL_ISF_REPOSITORY="<Your image path>"
  export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
  ```

  LOCAL_ISF_REGISTRY is your target container registry.

  LOCAL_ISF_REPOSITORY is the image path in which you want to mirror the images. You can choose your own repository paths. For example, hci-images/isf or hci-images. See the following sample values:

  ```
  export LOCAL_ISF_REGISTRY="registryhost.com:443"
  export LOCAL_ISF_REPOSITORY="mirror-fusion-services-images"
  export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
  ```

About this task

Points to note about this task:

- Run all commands as root user.
- In the commands, replace `<your enterprise registry>` with your enterprise registry.
- High level tasks to complete this procedure.
  1. Install `oc` CLI plugin. For steps to install, see Mirroring prerequisites.
  2. Set common variables used by all subsequent mirroring steps, such as the target registry and target registry location.
  3. Mirror IBM Fusion and related services (Backup & Restore, IBM Data Cataloging, and Content-Aware Storage). For steps, see Procedure.
     a. Set environment variables necessary for mirroring IBM Fusion and related services.
     b. `oc-mirror` on the generated files
- For the individual procedures, see the following topics:
  - Mirror images for Red Hat® operator. See Mirroring Red Hat operator images to enterprise registry.
  - IBM Fusion images. See Mirroring IBM Fusion images.
  - Mirror Global Data Platform images, see Mirroring IBM Storage Scale images.
  - Mirror Fusion Data Foundation images, see Fusion Data Foundation images.
  - Backup & Restore images. See Mirroring Backup & Restore images.
  - IBM Data Cataloging images. See Mirroring IBM Data Cataloging images.
  - CAS images. See Mirroring Content-Aware Storage (CAS) images.

Procedure

Mirror IBM Fusion and related Backup & Restore, IBM Data Cataloging, IBM Software Hub, and CAS services.

1. Run the following command to login to the Docker registry with your Red Hat enterprise credentials:

   ```
   docker login registry.redhat.io -u <Red Hat enterprise registry username> -p <Red Hat enterprise registry password>
   ```

   Log in to the IBM Entitled Container Registry using the IBM entitlement key:

   ```
   docker login cp.icr.io -u cp -p <your entitlement key>
   ```

   Important: Ensure you review the offline upgrade considerations before proceeding with upgrade mirroring.
   Set the following environment variables:

   ```
   export LOCAL_ISF_REGISTRY="<Your container registry host>:<port>"
   export LOCAL_ISF_REPOSITORY="<Your image path>"
   export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
   ```

   Note: Port is a non-mandatory value when setting the `LOCAL_ISF_REGISTRY` variable. You can ignore this if your enterprise registry is accessible and has a secure connection.
   Sample value for without port:

   ```
   export LOCAL_ISF_REGISTRY="registryhost.com"
   ```

   See the following sample values:

   ```
   export LOCAL_ISF_REGISTRY="registryhost.com:443"
   export LOCAL_ISF_REPOSITORY="fusion-mirror"
   ```

   `LOCAL_ISF_REGISTRY` is your entitlement registry.

   `LOCAL_ISF_REPOSITORY` is the image path in which you want to mirror the images. You can choose your own repository paths. For example, sds-images/isf or sds-images.

2. Run the command to login to the Docker registry with your enterprise registry credentials.

   ```
   docker login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
   ```

3. Ensure that the `redhat-oadp-operator` and `amq-streams` operator packages are present in your cluster.
   Note: If you did not mirror `redhat-oadp-operator` and `amq-streams` from the Red Hat packages previously, follow the steps that are provided in the Mirroring Red Hat operator images to enterprise registry.
   Note:

- Run the following command to get list of mirrored Red Hat packages available on your cluster.

  ```
  oc get packagemanifests | grep -i "Red Hat Operators"
  ```

- If you do not get **redhat-oadp-operator** and **amq-streams** operator packages from the previous step, then you must follow the step 1.
- Ensure that you also add existing packages along with new one in ImageSetConfiguration file. Otherwise, old packages can be lost from the Red Hat operator index image.

4. Create the image set configuration for .
imageset-config-fusion.yaml

```
cat << EOF > imageset-config-fusion.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v2alpha1
mirror:
  operators:
  - catalog: icr.io/cpopen/ibm-isf-cas-operator-
catalog@sha256:361a92013a2faf3e58a98fee85a8149caa80eb9b216f84b5384944dff6a7187f
    packages:
    - name: ibm-isf-cas-operator
      channels:
      - name: v1.0
      - name: v1.1
    full: true
  - catalog: icr.io/cpopen/ibm-usage-metering-operator-
catalog@sha256:a97cef95c73ac6dacea010818757b4ed9b7440f4c0ec4154a091f248c68d1c9e
    packages:
    - name: ibm-usage-metering-operator
      channels:
      - name: v1.0
    full: true
  - catalog: icr.io/cpopen/idp-server-operator-
catalog@sha256:1b93488e0bf5c421dace9bae65fd992695c1e63fa12e7b82f03cf852d3fa02ad
    packages:
    - name: guardian-dm-operator
      channels:
      - name: v2.0
    - name: guardian-dp-operator
      channels:
      - name: v2.0
    - name: guardian-mongo-operator
      channels:
      - name: v2.0
    - name: ibm-dataprotectionagent
      channels:
      - name: v2.0
    - name: ibm-dataprotectionserver
      channels:
      - name: v2.0
    - name: redis-operator
      channels:
      - name: v2.0
    full: true
  - catalog: icr.io/cpopen/isf-operator-software-
catalog@sha256:569b3f5158af370abe3dbe049c53be55d32f239a717ca74f1f5e544c697e8f21
    packages:
    - name: isf-operator
      channels:
      - name: v2.0
      minVersion: 2.13.0
      maxVersion: 2.13.0
  additionalImages:
  - name: cp.icr.io/cp/dcs/auth@sha256:6cb6e582d02fb017dbbc1c736f3409a08823fef56381747f13952bd8a613cabb
  - name: cp.icr.io/cp/dcs/connmgr-scheduler@sha256:cca24250b9bbfdfe8c6c85c613c5fa1a56d2b0fbbb0e48a4c56fc3f50ea338a2
  - name: cp.icr.io/cp/dcs/connmgr@sha256:cbefb44b2d448b36f90e4fca7973ab66bc1fbfd799eaa1c13b5185bf116f6f09
  - name: cp.icr.io/cp/dcs/db-schema@sha256:534c98417f909a248698281bc244ef1441b6a01dccf5a3055242c13956576c9a
  - name: cp.icr.io/cp/dcs/db2whrest@sha256:3233681ab2ec32515653f5659d6c02f8e12fbfddaa9d65747dfe07a62de1bfd2
  - name: cp.icr.io/cp/dcs/dcs-ai-mcp-server@sha256:d8af262ff80ee897797d322bbd8f8382cf4b823033901aea37d5531b27565365
  - name: cp.icr.io/cp/dcs/import-service@sha256:1003a8694eff91c163a72b88c19baeb5cd47a97d152ab64ac910b8b8b693f096
  - name: cp.icr.io/cp/dcs/import-tags-app@sha256:c37128ef0ad7f7add93c72c42c8c1902f75a2907e683af5d7af99d04ff4ea381
  - name: cp.icr.io/cp/dcs/isd-proxy@sha256:5056d6a265b2a37123806976c56722b86c5c20dad6d2f5933838c8d076b1e120
  - name: cp.icr.io/cp/dcs/keystone@sha256:b3128687e35f78bc25cefbe7254f625b5f01f344ae190d161cdbf79057109aa6
  - name: cp.icr.io/cp/dcs/llm-proxy@sha256:fce3aebc677eba45434f16039d3c678ccf4c1c71d709ab3b13b5aebd79c928b6
  - name: cp.icr.io/cp/dcs/metadata-extender@sha256:9a988643df596a1bbfb1094874273c3c1ca879e8a645677090409b4de1f37ac9
  - name: cp.icr.io/cp/dcs/mo-agent-extract@sha256:2f4d7b0d5ada12d39932f687cea031d88beb9cdac791b48cdca29ed23a9860ca
  - name: cp.icr.io/cp/dcs/mo-ubi-init@sha256:927bb277f576f2a772faf8dc9978a6592d5d08bfad3cdce81da180a9095a34c2
  - name: cp.icr.io/cp/dcs/moconsumer@sha256:2f1d3fdec099951cfc5f6788d4a4d944d66ec0aa8e39c9688239f1ecad38729f
  - name: cp.icr.io/cp/dcs/moproducer@sha256:c984c05bc47939e22cec6f4c8c0a9f365b71b7582c3711a36a59b96cc546da78
  - name: cp.icr.io/cp/dcs/must-gather@sha256:3f655e31efb04862c52333f113322a9169ea54d88f627177707453fd48acec50
  - name: cp.icr.io/cp/dcs/policyengine@sha256:ffe2fac7ee8bd49be48a142818ca760c6482018d9a79941296ef8d3424c1949c
  - name: cp.icr.io/cp/dcs/remote-content-search@sha256:7c0a39bcf04c1063fffc866ec8513066beb398dfd28cdb436b21cf7c4754f7c8
  - name: cp.icr.io/cp/dcs/reports@sha256:a95d112e43f1ecf7f962f04354c64be6052319eb0c5cabf94c9af325648d9f46
  - name: cp.icr.io/cp/dcs/scaleafmdatamover@sha256:19aa20ce2b493f420e03758313e02f0351cf72836c15905f3c4b7285603e29c0
  - name: cp.icr.io/cp/dcs/scaleilmdatamover@sha256:725614999df502d09ba0621c4a2c317ea18818cb288ad38d0c46f98c78648ba9
  - name: cp.icr.io/cp/dcs/sdmonitor@sha256:2d853b11e533a9eee9e922034478289ed82d4641a16e9c797fafbaaff8fb1fdf
  - name: cp.icr.io/cp/dcs/tikaserver@sha256:11a8175a621e6a5fb093ff41d71eb20f8dca8a4186a24a843cc2713f634b382d
  - name: cp.icr.io/cp/dcs/uibackend@sha256:1bfd41169c4450852872b0f8d190d1b24b12a46040eb6a167437a2f8086c798f
  - name: cp.icr.io/cp/dcs/uifrontend@sha256:58979c1c53c116e5f90e814166a129a1fa8587831f617147328b5b0a5843d3f1
  - name: cp.icr.io/cp/dcs/wkc-connector@sha256:3a68bf843669cce6f000ca14ecd5d42a6727fd8055de5ac157b9ecf360fd5c59
  - name: icr.io/cpopen/db2u-day2-operator@sha256:67e656ec5ad43ef62550852f42b346ee68b4f2e97867980e822819687aa740ba
  - name: icr.io/cpopen/db2u-day2-operator@sha256:6e87d78d96101ec3a8dfca7464e4196bd34b1bc700b7ef1b1de5add00a591e55
  - name: icr.io/cpopen/db2u-day2-operator@sha256:af106b46b923745688ef2952da39add7d38b46768e18bf2e2147b4b05e755f00
  - name: icr.io/cpopen/db2u-operator@sha256:9062cf50e0d1df382878bf3ad4d5c5cfdcdc801666e419cd3c6a62838445978c
  - name: icr.io/cpopen/db2u-operator@sha256:b838f643c872fee3552f889c7977d593438e6b9583b03798555f44737d601e82
  - name: icr.io/cpopen/db2u-operator@sha256:df5b2ee589a381253f8d0797885c8f798351914edf9e506cc1ff261ad3156f09
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:0f013b2188b1e44fd450b4753b66e38f0a3624cf7f0b1bdab01ab36105a5d08b
```

```
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:10e960e1e26d3b64ceda847618b757e1d7e162a3378668d1f85a14d1b38ec55a
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:9c3f363473b182a4291e2bf07c630bdc8907abb26b5ede6c36f55f3e07de6ac0
  - name: icr.io/cpopen/ibm-db2uoperator-catalog:2.13.0-
21838@sha256:3d8a379ecc97328eee2e4d5c83f380c87f1b5a32c4097112116f24f287226830
  - name: icr.io/cpopen/ibm-db2uoperator-catalog:2.13.0-
21838@sha256:d6b6fcd8be9359bc8854fb314a6a73b6b5018d3dda2a005a1d16f8b15b18d1a8
  - name: icr.io/cpopen/ibm-isf-cas-operator-catalog:2.13.0-
21838@sha256:361a92013a2faf3e58a98fee85a8149caa80eb9b216f84b5384944dff6a7187f
  - name: icr.io/cpopen/ibm-spectrum-discover-operator-
bundle@sha256:cced71d877089db30e7badc95a2113ae09136347777cae0ec49000a71beaac81
  - name: icr.io/cpopen/ibm-spectrum-discover-operator-catalog:2.13.0-
21838@sha256:c8b42eafa039542123c4ba29ef650c4a41996a28fd4504fb7c4e2e62fedab5b1
  - name: icr.io/cpopen/ibm-spectrum-discover-
operator@sha256:0b97164be106bf8733de8e432c17afc9e597251c64d0a2c37f2da5edab70f9f1
  - name: icr.io/cpopen/ibm-usage-metering-operator-catalog:2.13.0-
21838@sha256:a97cef95c73ac6dacea010818757b4ed9b7440f4c0ec4154a091f248c68d1c9e
  - name: icr.io/cpopen/idp-server-operator-catalog:2.13.0-
21838@sha256:1b93488e0bf5c421dace9bae65fd992695c1e63fa12e7b82f03cf852d3fa02ad
  - name: icr.io/cpopen/isf-operator-software-catalog:2.13.0-
21838@sha256:569b3f5158af370abe3dbe049c53be55d32f239a717ca74f1f5e544c697e8f21
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:1b0dab1db92c2a6c36ad24ac9d5aa52acde882dcbcff4e6f4ad8af4fa9b857b7
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:39baed72d1a73aa5a2fbf245d4fe20faa6762908acb6efe9659e6cc6bd3a5c6f
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:46e9621ee884f45907330610be5f5a4e4f5f978da2d35d4e0d77e70abd767386
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:4a34187ba2196012acfd2eab8fb06b07c679bc6c0c13fcc988676043afc1c968
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:67fb65c796d162a58648f2ede6a053e1b9fd76aab01c91aad9acf39c5afbb814
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:6a123d208d9936f9f0148588e944edc348da066570f11ba6627659acca79794a
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:9a9fcad406d1500f3b86f6820ca80b0fcf4ddc6b8e2b6796b60eb53e0c616a76
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:c93051ba7ba210c4ad1a4c904be3752355c5937daf10298a14fb02c098a24c71
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:fffb214cf777b845fa4a4617ff59a5b14d4ef94ca7e430b684bcbc3db6c75939
  - name: icr.io/db2u/db2u.db2wh.restricted.v2@sha256:6855ca6e06372a9072e248ccb1c5770b65400b1226859402411a9c18e0426789
  - name: icr.io/db2u/db2u.db2wh.restricted.v2@sha256:c6c98d491a01e10bf244e39b86d0073b3c3da4ccefaa15c15e9db562f7da0da7
  - name: icr.io/db2u/db2u.db2wh@sha256:0aa85bc1e6c8fe891fc4b3f2137d2afef611a52e11134055a76ce3d5602b6200
  - name: icr.io/db2u/db2u.db2wh@sha256:3cc0814e4850663815b34a21ba3d480bebe4d312031315c192ad33d1ef0d6b09
  - name: icr.io/db2u/db2u.db2wh@sha256:a21f1de1cf2accdd0c0e94f045cda34f1d2b6a0b292bce33e66fb0eb93fbf45b
  - name: icr.io/db2u/db2u.db2wh@sha256:da3dcced05f576ff140647b8fcf73bc2e8aad9d329cfb34d042350c487798caf
  - name: icr.io/db2u/db2u.dv.api@sha256:c775f04c1cf735e4eb9b654f8307e69a294d33a182310b2068c631365ace03e2
  - name: icr.io/db2u/db2u.dv.api@sha256:fe5d3b18469795b2bf9635a0d85b1b84542dc1506aaa5f3f5568620c1b11bc39
  - name: icr.io/db2u/db2u.dv.caching@sha256:16b671ca11da90cb2032f52e0adfdb11763137255b2f8abd75d0907c3c36ee6c
  - name: icr.io/db2u/db2u.dv.caching@sha256:52ab0a6385636334cd3f405b62aa5719c4b600e59ac1f2636977300c3d798297
  - name: icr.io/db2u/db2u.dv.utils@sha256:8327012b2fbe2789aee4855a54125dfd114572790d08054b205ae9f24d679587
  - name: icr.io/db2u/db2u.dv.utils@sha256:ed9da17c3676c69285729270d820b953be7b49074047ff94e61962e4fbd596b62
  - name: icr.io/db2u/db2u.hurricane@sha256:2e8c303ad6878d2de0b80c72f47140b93b25abdfe320aaf3e0de0380edf3ddb5
  - name: icr.io/db2u/db2u.hurricane@sha256:eb88a59e26bf023af2cf97c9fe162be817dcb2afc6a8fab053fc06989a243875
  - name: icr.io/db2u/db2u.instdb.restricted.v2@sha256:8f49d7dab7152d689d057297894174bd3e5ce604b5357d776223bc243fccfa84
  - name: icr.io/db2u/db2u.instdb.restricted.v2@sha256:b7112c671588766b31a889582ceef49557117c72a73ce0c94e53bd7739879e3c
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:1036e55662f721284a535bc63a46f1d8c61fa17ce2eb4c4b90f147f69ec35290
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:28982d3ea268ab42a865f96f6f93ad9e36602c4f8dc0a0cf87d29e0570bd7d40
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:56a9a4bde204ac715f6762704f263c6f384412d370a1bcfd629f91a06995eed6
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:751cdc1796099c74ced03bda277dd1950166255d4aa0410a7f21e94276c68139
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:76ab047ae55cc5f1d34c46cbab64e33c725d38f93a9d74c62eb5426521718a99
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:a2a2edd7329585d5788ddd3e02865ccc9f2f29a81430e11bb9278af4c5b9eff9
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:ae828eeb870bc285298c3647c8767de790e36763984702db9621e94c9985e17e
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:df1e5e85210883d5a4586eca57d57182cb5a0974cc23cb5b4cd66aea2a2e749b
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:f22f0ab6ba02dfbdb353241250e7c4bb2e834b6f7dd89c88abe502e96916ff15
  - name: icr.io/db2u/db2u.instdb@sha256:1679240cfce999a2c4b4cac5371ad2c08784223acafd009347837498bd60315f
  - name: icr.io/db2u/db2u.instdb@sha256:275ac3b1b86238c80b9ac66619d868a7ece1a2c556198f5b5fc59a9b8eb84dc3
  - name: icr.io/db2u/db2u.instdb@sha256:6a12e1dda62f990cef4fd71cd5586dc724a2bd9c494e16b6cae9c89043c9a43b
  - name: icr.io/db2u/db2u.instdb@sha256:6b4cede07d71895f4f1983c1e7c09fef75324ec05b0ed36febe1f64a1b4773cd
  - name: icr.io/db2u/db2u.instdb@sha256:969417d181b532fbdb6853bfd15dfd391b44685d540661c53441f54d7d1d8f0b
  - name: icr.io/db2u/db2u.instdb@sha256:bff81001e977b2e4817fd998ef35c862a2fcba3b3ef80d29c8fc5d905e02fe81
  - name: icr.io/db2u/db2u.instdb@sha256:d5641eebf385829def0e5c667a0694b982f5d7c6460600e70c865f6fafb18de2
  - name: icr.io/db2u/db2u.instdb@sha256:dcd21a12aa9406313619fa05833e80cd64a52baa902cdde825300bb1a2b69b75
  - name: icr.io/db2u/db2u.instdb@sha256:e65987d494390105e9a9982b915ef362cbe5ee289582ce101063785ed16f2d05
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:13cdb4161aba7c9d2ef65a1eefd1131fb661bf0edbb05d475186280bd2661768
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:16b0a2fe8046660194792392470ebb5a6a18e82d5852ee31fe2f71234f1712c
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:408e13e0a250bd7daa1fe68f293dd5d22153ebdce8be409257bb3fec8f4406c1
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:5e80225570f566487355adeb9ed20ee924d23f8719270a704f38956a1bbe8b9a
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:63d959334719870cf211e44897af2d05e5b5fb9d65daa4b86c48db51582a423ee
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:a9e3ff99fa605558085505b3ecaae9be3371ae17ece47db6edfcedb0b2dcf5
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:bff1c6828d113f0997ebad884fadcb3953653ec29714cf81643b6f669e190fc8
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:c46a7d08092cb9593f8f6473769d1de04533ea04fdadf3093fc51c59bca3500d
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:e1057188f633d8ccc6b8ca3c6830f3d71266f5998e5800f9fe572529d4778b77
  - name: icr.io/db2u/db2u.mustgather@sha256:01951471547ee1ce3190fc2c03ada0f47623b7928b184d22173d06aeb458d73b
  - name: icr.io/db2u/db2u.mustgather@sha256:09268f018b77b94e8536c0f1aabec5fc8f5610c14bf31bedc59e88cdc78177f2
  - name: icr.io/db2u/db2u.mustgather@sha256:11035b1812f9ae1792e5b45f0e4857d5914f638a6b060a66a0aa4fa5e1933fca
  - name: icr.io/db2u/db2u.mustgather@sha256:19f59de73fa2df4409ec57f0929f30820393f5ed1cfa6dc866fc2e7f3321ffe7
  - name: icr.io/db2u/db2u.mustgather@sha256:2a9e534ea4819b46d5fac2590251a07556155bade8aeb04ff5183e9f942ee31b
  - name: icr.io/db2u/db2u.mustgather@sha256:5a47466d9a0e43ea5c477fb66a84bd1d0f4e7f705ed2a5ff152730de6a7dbcaf
  - name: icr.io/db2u/db2u.mustgather@sha256:acbcca4773f79fe966a3685a24c9734481d2482e7c6137300b494b8bd12ab1cb
  - name: icr.io/db2u/db2u.mustgather@sha256:c100495d77c36755f882675600b48c39c3261609008d9512aa76a27397302ebd
  - name: icr.io/db2u/db2u.mustgather@sha256:db3d978cfa5553d0843e1320e9715b6d089c3e369007ed502459d2a1a2ba860d
  - name: icr.io/db2u/db2u.qrep@sha256:78b41c0f0ca28fd6b90b1aa26f7ed6089749d29c8c62d7698155dad7bc3582b5
  - name: icr.io/db2u/db2u.qrep@sha256:881cf716b9f2389fe59809d409f9f9f2ca687c6a665c0330edf208d34777e912
  - name: icr.io/db2u/db2u.qrep@sha256:a8069c11a1adcd3934a9ba3cdc596308ecdc84c301a735c6506287f398d01fbc
  - name: icr.io/db2u/db2u.qrep@sha256:c08a2f9d4ea71b44d0444b1f61ea898ba1d67fed2cf0023cf774dfed095107ae
  - name: icr.io/db2u/db2u.qrep@sha256:e14893f0d00a8d06be5a9476dd1f3451957c454b7567fa30bfba4d9432b6a34d
  - name: icr.io/db2u/db2u.qrep@sha256:ed535873bb8606cef4bc6d87d036268647ab303e45a5849b3c3d03226b7dc6fb
  - name: icr.io/db2u/db2u.rest@sha256:0c41319691497cb7793cd91b53bb3e7084ea02ece9589d830ba5d130852125cc
  - name: icr.io/db2u/db2u.rest@sha256:1352b0fb4550dd13e4bb08e58832f6c203eba993950166dd9f003a9e8c162432
  - name: icr.io/db2u/db2u.rest@sha256:209edf7fa0c155efd7e4c209d68abed1747f8a02ba340f7015f0ee436da7ebd7
  - name: icr.io/db2u/db2u.rest@sha256:5cb5f6e25c4f6a789d1680478c266affe7719964faa16328822621c5f43b0cce
  - name: icr.io/db2u/db2u.rest@sha256:681ace912b6df4732c6b364f585a6782fcacb102e0465cb0df44e7417ee6a9f0
  - name: icr.io/db2u/db2u.rest@sha256:cd57827e98bca2488dc991c65276fe125e3a32f5c2fa95df1151e3deb9bf412f
  - name: icr.io/db2u/db2u.rest@sha256:d04710c5df92796e864da8cdc403b9ee2fa8bf8f006d6cc8f0cb59340f8163c3
  - name: icr.io/db2u/db2u.rest@sha256:dc657410c33beef847ffadc8e47f897fdf3c614800cddc7c52e48dcb7cacea09
  - name: icr.io/db2u/db2u.rest@sha256:f2e19e2a144c7e852baf4150e0668bdef37b9ea30c1ca7da773467ed780fb5f6
```

```
  - name: icr.io/db2u/db2u.restricted.v2@sha256:1788f4876b7153f51b28ca9307748d0f83ce4e42cd55ff01e29f76adc6bb29eb
  - name: icr.io/db2u/db2u.restricted.v2@sha256:df99b606a2b1683b2b064f1191e651f1df2c64458169de8f4cd14e1d10805876
  - name: icr.io/db2u/db2u.restricted@sha256:0c5b47350e0189147104bc5dd273803868608ab048f78783c6f640f4419e89eb
  - name: icr.io/db2u/db2u.restricted@sha256:20a17afd17d48c80b6129b8bcbfce7117a5fee909c44d27603cd6c2807b2dfc8
  - name: icr.io/db2u/db2u.restricted@sha256:565ddd470867984d282b26873c1440a0dd96b664a1f4f169b17a1b76065dadf8
  - name: icr.io/db2u/db2u.restricted@sha256:6999dba7f8c1c348dc79d81bbfcc468096db6d3ef7acb01facd3b60108807d9d
  - name: icr.io/db2u/db2u.restricted@sha256:7698c04b23be8058facc1d2dee69b6a94d9f95db9e25de3eafd9fac2622f8d25
  - name: icr.io/db2u/db2u.restricted@sha256:8a207ff2da1a29cae1c9d7ef521ca27a3f30330d43a7c351a379f93e178287ea
  - name: icr.io/db2u/db2u.restricted@sha256:8f89f5476a4d42ed558a8d5fe39abb24c75152a79079fe6a462010017319d5b6
  - name: icr.io/db2u/db2u.restricted@sha256:9d73c208aca0a993487289905ce7932cfc46ab0b802aac213aeb0cb16ad44e80
  - name: icr.io/db2u/db2u.restricted@sha256:cae02720f1ed588e416e24c3d3af74253fd2b29a5160bf699e5db1943233aa25
  - name: icr.io/db2u/db2u.tools.restricted.v2@sha256:4dc23d46e8a535661e54ee57f9147b1dc21eef6fa70e07e0b009fcd932a3d505
  - name: icr.io/db2u/db2u.tools.restricted.v2@sha256:78ab6edca7f3f725f1dc58d9010b0ccd3483df03ad8920ec7464c9c426e522b4
  - name: icr.io/db2u/db2u.tools@sha256:126301cb3616037133324c8c1352af0a7921226203a2ee9b9eeb736872e03311
  - name: icr.io/db2u/db2u.tools@sha256:22e995c30dae2dd792fcd0e07a7cf2b90fee23b67ac90b0722f11093c114e52a
  - name: icr.io/db2u/db2u.tools@sha256:38c70063a9700752b59d216a7ff7cfdb38fe4f40e345a091bd591feffe0ea2a5
  - name: icr.io/db2u/db2u.tools@sha256:67130a4444d5e5d5c9c2051f4983f0033a3a329aa1cb2e2488c06360f7134554
  - name: icr.io/db2u/db2u.tools@sha256:7e9f29fff47c0d8a7f4fe8feb4763cc3df5aa71b761434ccbd43d5a30fdaae3a
  - name: icr.io/db2u/db2u.tools@sha256:82ffd808fd4f316b6c4cbb327ab358cb0ceb1c110f12dd8a44ecc0264a33328d
  - name: icr.io/db2u/db2u.tools@sha256:b48f52abddec7da941c1613010936f9f651b70739194198914953cbb80992dae
  - name: icr.io/db2u/db2u.tools@sha256:b5815e23b5b60d4ee273ea52f4b752d1dc913464a29f59daee1eefd67cb6e87b
  - name: icr.io/db2u/db2u.tools@sha256:ba91e1d8780638b9871214449b8d76b2851c9d6e8c389990861df3669e2b198d
  - name: icr.io/db2u/db2u.update.image@sha256:0a0cc2a32b6ddff104722696b3799e43e94861c1f6af2d5c545b92aedf21cad5
  - name: icr.io/db2u/db2u.update.image@sha256:1a3c1e315216aee550aaa34f58b6f62c4ce274424afe3e5f569f083badbb5619
  - name: icr.io/db2u/db2u.update.image@sha256:2f4cb0440157f2199d6a0f73375dfa70953d2d22cd731984ddec761af2ed75b7
  - name: icr.io/db2u/db2u.update.image@sha256:5ba422423bc4ec0ab4bf6e044038fb71e41f51b9bd958b40aa4c54b807a6d45b
  - name: icr.io/db2u/db2u.update.image@sha256:770b560ba8e16a0be7b07edc11c41fd46ed0ad849059199fea207569d49ef953
  - name: icr.io/db2u/db2u.update.image@sha256:83a9f12d93fba82b7015819fd0baffa68e83644cf903c3df0d0c535141a8669e
  - name: icr.io/db2u/db2u.update.image@sha256:8e547fc1968b187a28187a4b998f3f70ebdeeb0aefc4938ee058db8588eae421
  - name: icr.io/db2u/db2u.update.image@sha256:a8ec7ac89729d73195459bb6a55b7151f4edb66c93fc4ad3aaf51c942116629d
  - name: icr.io/db2u/db2u.update.image@sha256:bdb4840d7fb72a69471da736704fa6968341182b5447ebb31a5c6a9223ac440b
  - name: icr.io/db2u/db2u.veleroplugin@sha256:2891904a3dcf7fbdcddcf48fe78bc00a7f31e2604716e99011114bb3d1ccf40d
  - name: icr.io/db2u/db2u.veleroplugin@sha256:4096b0785d1fa5019dea77017c47fb0d4ad32fceef2ffc23c88095c630bb6d45
  - name: icr.io/db2u/db2u.veleroplugin@sha256:6905ab5396f6b86c0eb70513b9b5f871c4d3832efee7694e8187451b4d8115a4
  - name: icr.io/db2u/db2u.veleroplugin@sha256:6e3141301930950e3a0671dd8d74a7148a7ef790284e607fd3d6870edcbd823a
  - name: icr.io/db2u/db2u.veleroplugin@sha256:73d8c40d160da82b2218b40ac5e1b065339db8d7bf6914ac694ebba071a858ad
  - name: icr.io/db2u/db2u.veleroplugin@sha256:7a890053b98b6e510f9697175071bc5bae00cdd1ea671c2949dfec9a61830403
  - name: icr.io/db2u/db2u.veleroplugin@sha256:ce73ee327f940e12bc22e20324d72acd7266f4adf730469d4fbbec482481f8e3
  - name: icr.io/db2u/db2u.veleroplugin@sha256:d327a193a266c982c30bc009a8def0c84d75dd4f28bf3806bee82e8b173946a0
  - name: icr.io/db2u/db2u.veleroplugin@sha256:f4f2e9e3afdcaa7a9e4e522bcdb2aaa6b79db97361968bdf1428961b1ada878d
  - name: icr.io/db2u/db2u.watsonquery@sha256:33d6ee3007e62284cf7852cfd337393a0bfb86432df599fd971c31836abaa28b
  - name: icr.io/db2u/db2u.watsonquery@sha256:48cce86af50ba003dbed9ce54b0240b9eec59a2702ab93bae58a649d7fe777ce
  - name: icr.io/db2u/db2u@sha256:0ecd06074f6ce0f090f93b2b5ed51e41679154d638dc2031d64498aab2c0de68
  - name: icr.io/db2u/db2u@sha256:323e043a92970faa921733dbd4d474bab3e80dc105fc6fe4b9c5e0d2160e1856
  - name: icr.io/db2u/db2u@sha256:64d1fc881fa7c4f9203a1e43b498464333768623cc1494d42e2d8aac21b4af3c
  - name: icr.io/db2u/db2u@sha256:9678623a716ceb4ed9367812b842cd56227ed15858edae08f7bcd5961dfa5e2d
  - name: icr.io/db2u/db2u@sha256:b8701f8d719345ee156c2490d8a0fc9ee08a2cfca7c4537d728c729337da9b77
  - name: icr.io/db2u/db2u@sha256:d282a83ba1d016035d9e7c8f489b3599cc3e39826890e927856a5232b6f6013a
  - name: icr.io/db2u/db2u@sha256:d7c92b368064d15ddf6bcc3d96dc19f876603f5a9fb453185724607e2b966567
  - name: icr.io/db2u/db2u@sha256:ef563ca49bf4ad388e8af360a5a1d87fd756d0e9329c6a181f3a37c29429b9c6
  - name: icr.io/db2u/db2u@sha256:f95b4ad43756f1f256df6f9b8cc79757db430de06a0fea22b78261490a73649a
  - name: icr.io/db2u/etcd@sha256:3c5d8ff1c77f146b6db7224bcfc92d6d64a832332cc087a162b2cda81ea99eef
  - name: icr.io/db2u/etcd@sha256:84dda9076ae58f2f02abba6102ab8325e4c301984b2588cb751e46245c636940
  - name: icr.io/db2u/etcd@sha256:9fa13ffeab820c5a62bc5bdb41347ef5cfd5244584117f0f3bd28cfe8690d358
  - name: icr.io/db2u/etcd@sha256:a5123f1a68e08bbda547eb653b724b917c9ec4d4cb6f447b066c4b71c68280d9
  - name: icr.io/db2u/etcd@sha256:af1d052b60f599452ecbb44a060173922286e296cbf52d5a551a485cc28900e6
  - name: icr.io/db2u/etcd@sha256:b211cdc9c094ecad0f8bb3d0b714331937bb26582e450b3babec158f9857976a
  - name: icr.io/db2u/etcd@sha256:b617959f55b413b22a5bb72a80c19c7ff966a695dd4793903c9044165296aa3c
  - name: icr.io/db2u/etcd@sha256:cc1f51f48d7d95bd25a1658490adbdc0b7d8d0429e23ff621b3b8b457d51e271
  - name: icr.io/db2u/etcd@sha256:e386ff2ca65b99c63c6166de9a649875891f493091754aa4dc0e83e92dae7cb2
  blockedImages:
  - name: icr.io/cpopen/db2u-day2-operator@sha256:91a7ebbbf0667cf18e6686145dcb9acf2df2d616ee610733091002 47e3b7e6c3
  - name: icr.io/cpopen/db2u-operator@sha256:4a938eb64e44e5bf155293e1939c4c8f2bcee9603647167994a94dbd79ced316
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:8fc35a4bdb7db8231f0258986a3dcad014b80059681a55ce4cd2a87659c52849
  - name: icr.io/cpopen/ibm-db2uoperator-catalog:2.13.0-
21838@sha256:3325c4e4ec118c186f80b8336e5a3a5d8e2c2d7a1f06d97c9fee1cc4c90d68dc
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:12713664b4753551ae3ac303c25c4df938a4394e241e1f9bbc526e74a8ad562e
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:4b890828cc50851a42e5fec6bb929d1f71578aeb143794cbfaa1207eb0d7d048
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:890d04affc2beb381a483162aa3e998599b308ca1615ecd63e66ba04d94c224a
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:b711c164a62ccd4db955ea67afb0e225601429b6406d98fbad31c5fea13a86ea
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:d35b06c1d03707642c6641aff7d481be128064b3aabb384e5aac43715da45d04
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:e57279502a5d605931494771e8de0b382d3ddb1a5d744820f70977bb777bdaf9
  - name: icr.io/db2u/db2u.db2wh@sha256:1594796ebbe52b158c12eeb8e5d4faa791f6c9a28f96e33844a05f0282b4fc1f
  - name: icr.io/db2u/db2u.db2wh@sha256:394a2a0c640a884c1f63332bce1a8d5ad73baba0d3b3a05b4fa876e35bf9d82e
  - name: icr.io/db2u/db2u.dv.api@sha256:f8ed8a89397c5aa2d6b61ac2ed9d55e3b6351c3c0396355c17d7e44a48038def
  - name: icr.io/db2u/db2u.dv.caching@sha256:d5fe8bbd87c5d4e4424f5bd71ae3bf35769ea744322b152128ab563412eb68b0
  - name: icr.io/db2u/db2u.dv.utils@sha256:83ec4f59d30685a3ab43c74ee20336a16635935f339b78c4744058504e110f5e
  - name: icr.io/db2u/db2u.hurricane@sha256:7e7d7b1df0122120aeb33864b00cd5b9d8a2beab0ee6e5bbe273b0b7ea4f0e14
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:22bb255ac34cff1d99255a52a94b6560880f5f51937fdfcedba1ccd668688d9c
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:61c8a176b449e0aa7638a524297a80ece4ec52700d99299ae855db5b54609438
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:7906196d7910eada7f94bb375f4d6d7e5174992a6a0b3aa3308ce12283df45a8
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:7ea95db7c6249ad9b115c5934b4849f68b633024db937bb7835ec3cb4ab6a897
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:b67acd5ee75833ec1e10705a0c4e7ba0f4be85ab03ed37527b56250ee3448cb9
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:d5e452c1f3f5dcacb747713e56dcd006be6ba291bbf305352cd411a9bae57ee6
  - name: icr.io/db2u/db2u.instdb@sha256:4e8688e4964cbbeb4a5329294a0d995a0a45a9bec237e3f0a18491f0bce97784
  - name: icr.io/db2u/db2u.instdb@sha256:5ad3fc61b531781b212275eed3c3632ba0b7cd7d5cb48cf321b3f2308b66c421
  - name: icr.io/db2u/db2u.instdb@sha256:829a95722e17914869a4e224f35ace529607e71a1d2235f6e6fe7f33378d1cb5
  - name: icr.io/db2u/db2u.instdb@sha256:a6c008e42a208fe1407cf8e9a9c5e4d97eb923aee7269821fad9bd037b407357
  - name: icr.io/db2u/db2u.instdb@sha256:cc300234190d53a01cbe7349b628df0518553b4cd6795bebfbbacd66ca5ae29b
  - name: icr.io/db2u/db2u.instdb@sha256:d1f0a4cd8591bc509208ab01d0268881d4a8529d53bae147e570de5a0ead3bb1
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:344840dc5a825a755f9701792bfc5f8e7766b776665c88e83a04bec8a1cd85ac
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:7313133d3b83dbf4a214c73e809b21b431bf81f4ff3c027a4ebe0608c254c9fc
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:7e7ac8e3f66e45c99e4fcf8673e09b18e2c0f7a0d6f72c3f86eb2eb157fabaf8
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:88574364d46b83ca98f20569e7ba0533fd38dd6a3cdff23bc374a629de1379e3
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:b8288381e8b2538464097db4328641648a44c5e9671bbc502af2e997b39abb41
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:f84b460effa534c060592012f01d85f16d340557931e3532f5f6e0b4b892b266
```

```
  - name: icr.io/db2u/db2u.mustgather@sha256:51707eef34bad0eeeecfb09c5746782851a0b4d6af562b626dc667ec5e50c723
  - name: icr.io/db2u/db2u.mustgather@sha256:517ae4be20211a68c11d5848f6330ec156bff90c3c64057d273b52d4bb98dae9
  - name: icr.io/db2u/db2u.mustgather@sha256:536bf34b07defd3bc624a7a3973e6d5e701d5fdb949978cfcb1dbd546968c475
  - name: icr.io/db2u/db2u.mustgather@sha256:628ac4c7d5e9b6f527cf451f397da5a8723956a283fa9a3dfe1db2dfc91bc858
  - name: icr.io/db2u/db2u.mustgather@sha256:6c91db906f8dd87bef288595ef2393d910f124c8d31440731af8b260fd2fa55d
  - name: icr.io/db2u/db2u.mustgather@sha256:f0a3a3dd13c7d133d39b25234977a150993334ae45049cb51a7fffc15943cd9a
  - name: icr.io/db2u/db2u.qrep@sha256:3c731b912f43a68b8ab8fef2bd61a4fe5287fac5c251046a160a049730f60f5f
  - name: icr.io/db2u/db2u.qrep@sha256:44724bda2d5336f17bec950e40195f4cb6199291a22a3c6d3d9b47407e0155f1
  - name: icr.io/db2u/db2u.qrep@sha256:9512f41becbcfc6ebf8afcaf35521478fd87e2b2881d856d443b5bb882cb8d79
  - name: icr.io/db2u/db2u.qrep@sha256:d6bcb17bd3ff2776ee3588ddebd946cade42a76f38666ccfc425635b09a6eb12
  - name: icr.io/db2u/db2u.rest@sha256:6e1f90125ad8a4740bf388d16f44e737373379f4dd9b881bc2559aea2d60d9976
  - name: icr.io/db2u/db2u.rest@sha256:778a3bd330aa7c9aa7cf405551856e92d3d8b40d75a509054d8c115e73e7b795
  - name: icr.io/db2u/db2u.rest@sha256:a12723d1671e2321286c22cfad30352c7860cd95f8099e45b4847036ccc0830e
  - name: icr.io/db2u/db2u.rest@sha256:a8158630c0ad0a21592a9e08891035d0d02d7b406c15f4740db77fec125b03ce
  - name: icr.io/db2u/db2u.rest@sha256:bf6576eecd916419d2d521289970094e0c73ded5de5154030678d6543c60a2de
  - name: icr.io/db2u/db2u.rest@sha256:d000f87783bf8963423064de28d069e6877aed8ec94d572c6b22a27313fec8ff
  - name: icr.io/db2u/db2u.restricted@sha256:27e1777bf955d26ce713cb8d6ffc6ee1d8d76c79dcbe94c07d1ec154b12519bd
  - name: icr.io/db2u/db2u.restricted@sha256:33a70ec4786221efe5e5fd397951111fc90ea3b75842ef9558c889034564e91b
  - name: icr.io/db2u/db2u.restricted@sha256:4e4397c45c10f09f723ff11b3b51bb7adb492898d652501abd85511cc660ccc6
  - name: icr.io/db2u/db2u.restricted@sha256:6b204a6026e176bc7c7404fd41ede9399439cf5347698c95b838a78454ace6d9
  - name: icr.io/db2u/db2u.restricted@sha256:9b388eab7b17d543bf9c731a2583bd2606ba6199873ab914a3cdd2065e231385
  - name: icr.io/db2u/db2u.restricted@sha256:c75c667f44a6ab63447d0e72ad2743beb6f34f29f0cdee9b83d83c72ebb2457e
  - name: icr.io/db2u/db2u.tools@sha256:02f1771f4a00db39a5ef741dc39eb84caedfd7393092607994500d9b481b9a54
  - name: icr.io/db2u/db2u.tools@sha256:35ae6c6618b443c49a17ff32f5ba0c098e46fc542482c02e94857eb378ace419
  - name: icr.io/db2u/db2u.tools@sha256:83c657db8954d116a97daae90957ed8b31a89055913de08c92bbe5d30d3c71ec
  - name: icr.io/db2u/db2u.tools@sha256:adeffc101962a8181b4ceba587d046808c6440ca4a516cabdb04f973b5e299e9
  - name: icr.io/db2u/db2u.tools@sha256:e89e56844720d8512e0e0ff65a0331719bc4c203cabf2eb1da329a6dd3d00c70
  - name: icr.io/db2u/db2u.tools@sha256:f22e5bbc28263a51aa45c4a42dfaabee793322594c32a58a13914b977aca300f
  - name: icr.io/db2u/db2u.update.image@sha256:318c6f072f745c191feecb34576bb68681dd9db3644cc32191fba34274332852
  - name: icr.io/db2u/db2u.update.image@sha256:56ec63418fba021fb1eafe54bc6d62106dfa9819a3ce88707fabcadb98b239aa
  - name: icr.io/db2u/db2u.update.image@sha256:5af408ae4cb77592ee6726abe3a281b9a7fac500231d05b45b0a2b05ed88293d
  - name: icr.io/db2u/db2u.update.image@sha256:836a33719793c32744c0f830890633cd1648f282585ae672d3067bd146b1d467
  - name: icr.io/db2u/db2u.update.image@sha256:9a5875da2221ed1e2725a085ec7abd2e0418379686e337987f9496367e7c7a61
  - name: icr.io/db2u/db2u.update.image@sha256:d327bb0e5cc2a453d2fad23b629ba4862b5af9f794fc7f10ecf9d46e5d107f19
  - name: icr.io/db2u/db2u.veleroplugin@sha256:1a2d4657bf14fa60ffa1b0dbca50b696f1ff4ffd28f4c470db67b0ecca5d6fb4
  - name: icr.io/db2u/db2u.veleroplugin@sha256:51114e7afaa01a9002289017a22a2f5c9c0eceeb8f2df857c8148dbb5651e46b
  - name: icr.io/db2u/db2u.veleroplugin@sha256:796518c555504dd1e5b77504ca07397c114ac6753adbf8c8de41dcd2e1b57065
  - name: icr.io/db2u/db2u.veleroplugin@sha256:a2b0b6ff986b618fe880acec97d4af1673f4bffda385e05b6e14301c1b51d472
  - name: icr.io/db2u/db2u.veleroplugin@sha256:b3ecbc09923c288fa8d21d9e36a23a61cbec71a0c43bdb1e8213688bab571626
  - name: icr.io/db2u/db2u.veleroplugin@sha256:f87644162951b7210e450f567dacf718d3480a2f3468b36131b7a43286a6c44f
  - name: icr.io/db2u/db2u.watsonquery@sha256:897237498f224723b5818c50cd37d218032c4a05ebe2bda15bebe99d13ef0521
  - name: icr.io/db2u/db2u@sha256:01bdc550ea7d727fa5715f96c61f7ffe763ee38fc1875132187c40e09d08a6aa
  - name: icr.io/db2u/db2u@sha256:427657ba472b22f66b68d2a4f2ea7564572853083508ae2a8dbf770d1a181b85
  - name: icr.io/db2u/db2u@sha256:4319988ff6a8dff6a97368f253fed5cb268eb0c20de4db9dedd77cc47214f806
  - name: icr.io/db2u/db2u@sha256:6dda7e511b92ffc83c4b92e7b5ed99b5b619a5986af4a86d75c5f4425917dd91
  - name: icr.io/db2u/db2u@sha256:94226fa2a0fa69f8438265f481ffc05e830a75b9bb4a04f2a2b1f8d239758b09
  - name: icr.io/db2u/db2u@sha256:adc37d1fdfde105a516b8b891c494627703d90e9104181b2959a98e6630a4e94
  - name: icr.io/db2u/etcd@sha256:0c79ec0ad02a7e810ce4bc094fe85ad869ac973494161e85dbf32e394fcccf6c
  - name: icr.io/db2u/etcd@sha256:1a3362799bd9402aba7f99eed179c2747a520b37c994e63ad9e2b5fdeb4d5bd9
  - name: icr.io/db2u/etcd@sha256:1befb2b817dc6f7bf3f2fe3f5f2862b638d791d1284facddf789f245775e575c
  - name: icr.io/db2u/etcd@sha256:234c073c178f60ff8e8e29fdfda25b1e908ea0655ff61f3a4e29a51eefdea8eb
  - name: icr.io/db2u/etcd@sha256:6e0c097d8e6d9cdacc71f80312dd96b8a0d50a07ad48a333fce0bfc190f58553
  - name: icr.io/db2u/etcd@sha256:bb5c436b88c57715ffc991d0ab5a05979a4c092c65ad98296af2b1349926c45c
EOF
```

5. Run the following oc-mirror command to mirror the images.

```
oc mirror --v2 --config imageset-config-fusion.yaml --workspace file://./ docker://${TARGET_PATH} --dest-tls-verify=false
```

6. Apply the IDMS file to your cluster:
   Note: Replace the variable **$TARGET_PATH** with your registry details where images are mirrored.

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: ibm-spectrum-fusion
spec:
  imageDigestMirrors:
  - mirrors:
    - $TARGET_PATH/cp
    source: cp.icr.io/cp
  - mirrors:
    - $TARGET_PATH/cpopen
    source: icr.io/cpopen
  - mirrors:
    - $TARGET_PATH/db2u
    source: icr.io/db2u
  - mirrors:
    - $TARGET_PATH/minio
    source: quay.io/minio
  - mirrors:
    - $TARGET_PATH/openshift4
    source: registry.redhat.io/openshift4
```

7. Create a **CatalogSource**
   Note: Replace the variable **$TARGET_PATH** with your registry details where images are mirrored.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: isf-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-spectrum-fusion-sds-2.13.0
  publisher: IBM
```

```
  image: ${TARGET_PATH}/cpopen/isf-operator-software-
catalog@sha256:569b3f5158af370abe3dbe049c53be55d32f239a717ca74f1f5e544c697e8f21
  sourceType: grpc
```

8. If you planned to install the IBM Data Cataloging service, then apply the following catalog source.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-spectrum-discover-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-spectrum-discover-253.0.0
  publisher: IBM
  image: ${TARGET_PATH}/cpopen/ibm-spectrum-discover-operator-
catalog@sha256:c8b42eafa039542123c4ba29ef650c4a41996a28fd4504fb7c4e2e62fedab5b1
  sourceType: grpc
```

9. If you planned to install the Content-Aware Storage service, then apply the following catalog source.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cas-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-cas-1.1.5-linux-amd64
  publisher: IBM
  image: ${TARGET_PATH}/cpopen/ibm-isf-cas-operator-
catalog@sha256:182acd92bd262539c5cc13470465719cf56398188bc6dfa230ccec1b34d8ecf7
  sourceType: grpc
  grpcPodConfig:
    nodeSelector:
      kubernetes.io/arch: amd64
```

Important: The process also creates a CAS `CatalogSource`, which you must not delete. Running the delete command does not remove the CAS `CatalogSource`.

10. Hosted Control Plane upgrade:
    a. To upgrade Hosted Control Planes, mirror the IBM Fusion image of the upgradeable version. For more information, see Mirroring IBM Fusion images.
    Note: Verify whether mirror of IBM Fusion images is added in the `HostedCluster` custom resource (CR).
11.

Mirroring OpenShift Container Platform release images to enterprise registry for Hosted Control Plane

Mirror the OpenShift® Container Platform image repository to your registry to use during installation or upgrade of the Hosted Control Plane.

Before you begin

1. Before you mirror, go through the prerequisites. See Mirroring prerequisites.
2. Use either podman or docker commands to login to the registries. The examples in the procedure used docker; if you are using podman, substitute `podman` for `docker` in the commands.

Procedure

1. Set the following environment variables:

```
export LOCAL_ISF_REGISTRY="<Your enterprise registry host>:<port>"
export LOCAL_ISF_REPOSITORY="<Your image path>"
export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
export OCP_VERSION="<your selected OCP version>
export OCP_MIN_VERSION="<your selected  4.x.y-build>"
export OCP_MAX_VERSION="<your selected 4.x.y-build>"
echo "$TARGET_PATH"
```

2. Run the command to login to the Docker registry with your enterprise registry credentials.

```
podman login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
```

3. Create an image set configuration file for the OpenShift Container Platform.

```
cat << EOF > hcp-ocp-images-imageset.yaml
apiVersion: mirror.openshift.io/v2alpha1
kind: ImageSetConfiguration
mirror:
  platform:
    channels:
      - name: stable-$OCP_VERSION
        minVersion: $OCP_MIN_VERSION
        maxVersion: $OCP_MAX_VERSION
        type: ocp
    kubeVirtContainer: true
    graph: true
EOF
```

For example:

```
apiVersion: mirror.openshift.io/v2alpha1
kind: ImageSetConfiguration
mirror:
  platform:
    channels:
      - name: stable-4.18
        minVersion: 4.18.22
        maxVersion: 4.18.22
        type: ocp
    kubeVirtContainer: true
    graph: true
```

Note: In case of Bare Metal Hosted Control Plane deployment, remove `kubeVirtContainer: true` flag.

4. Run the command to create catalog mirror images:

```
oc-mirror --v2 --config hcp-ocp-images-imageset.yaml --workspace file://mirror-file docker://$TARGET_PATH
```

It might take 10 -15 minutes to complete.
Example output:

```
[INFO]    : === Results ===
[INFO]    :   release images mirrored successfully
[INFO]    : mirror-file/working-dir/cluster-resources/idms-oc-mirror.yaml file created
#…
```

5. Upon successful execution, apply the generated IDMS configuration to the cluster.
Example IDMS:

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: idms-release-0
spec:
  imageDigestMirrors:
  - mirrors:
    - <target path>/openshift/release
    source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
  - mirrors:
    - <target path>/openshift/release-images
    source: quay.io/openshift-release-dev/ocp-release
```

Mirroring Red Hat operator images to enterprise registry

Mirror the Red Hat® operator images to your enterprise registry.

Before you begin

- Before you mirror, go through the prerequisites. See Mirroring prerequisites.
- Use either podman or docker commands to login to the registries. The examples in the procedure used docker; if you are using podman, substitute `podman` for `docker` in the commands.
- Ensure that you go through the *Before you begin* section and *About the task* section of End-to-end mirroring of IBM Fusion and its services images to the enterprise registry. For more information about the installation of Red Hat operator images, see Using Operator Lifecycle Manager on restricted networks.
- Ensure that you install the `oc-mirror OpenShift` CLI plug-in. For more information about `oc-mirror OpenShift` CLI plug-in installation, see Installing the oc-mirror OpenShift CLI plugin.
- Ensure that you must use the recent enterprise registry images for mirroring.

Procedure

1. Log in to quay.io and run the following command to login to the Docker registry with your Red Hat enterprise credentials:

```
podman login registry.redhat.io -u <Red Hat enterprise registry username> -p <Red Hat enterprise registry password>
```

Set the following environment variables:

```
export LOCAL_ISF_REGISTRY="<Your enterprise registry host>:<port>"
export LOCAL_ISF_REPOSITORY="<Your image path>"
export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
echo "$TARGET_PATH"
```

Note:
- Port is a non-mandatory value when you set the `LOCAL_ISF_REGISTRY` variable. You can ignore this if your enterprise registry is accessible and has a secure connection.

Sample value for without port:

```
export LOCAL_ISF_REGISTRY="registryhost.com"
```

See the following sample values:

```
export LOCAL_ISF_REGISTRY="registryhost.com:443"
export LOCAL_ISF_REPOSITORY="fusion-mirror"
```

`LOCAL_ISF_REGISTRY` is your entitlement registry.

LOCAL_ISF_REPOSITORY is the image path in which you want to mirror the images. You can choose your own repository paths. For example, sds-images/isf or sds-images

2. Run the command to login to the Docker registry with your enterprise registry credentials.

```
podman login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
```

3. Create an image set configuration file for Red Hat packages that are required for IBM Fusion installation.
For example:

```
cat << EOF > imageset-config-redhatoperator.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v2alpha1
mirror:
  operators:
  - catalog: registry.redhat.io/redhat/certified-operator-index:v$OCP_VERSION
    packages:
    - name: cloudnative-pg
  - catalog: registry.redhat.io/redhat/redhat-operator-index:v$OCP_VERSION
    packages:
    - name: kubernetes-nmstate-operator
    - name: redhat-oadp-operator
      channels:
      - name: stable-1.4
        minVersion: "1.4.2-0"
      - name: stable-1.5
        minVersion: "1.5.0-0"
    - name: amq-streams
      channels:
      - name: stable
        minVersion: "3.0.0-0"
    - name: nfd
    - name: node-maintenance-operator
    - name: rhods-operator
    - name: openshift-cert-manager-operator
    - name: kubevirt-hyperconverged
    - name: metallb-operator
    - name: multicluster-engine
    - name: local-storage-operator
    - name: lvms-operator
    - name: kernel-module-management
    - name: openshift-gitops-operator
    - name: submariner
    - name: advanced-cluster-management
  additionalImages:
  - name: registry.redhat.io/ubi9/ubi:latest
  - name: registry.redhat.io/ubi9/ubi-minimal:latest
  - name: registry.redhat.io/ubi9/ubi-micro:latest
EOF
```

Note:
- The self-node-remediation operator is used to enable automatic workload failover in case of unplanned node unavailability.
- The redhat-oadp-operator and amq-streams packages are required to install the Red Hat OADP operator and AMQ Streams operator. It is a prerequisite to deploy the IBM Backup & Restore service. If you plan to use the IBM Backup & Restore service, retain it in the commands, or else you can skip it.
- The local-storage-operator and lvms-operator packages are required to install the IBM Fusion Data Foundation service. If you plan to use the IBM Fusion Data Foundation service, retain it in the commands, or else you can skip it.
- The amq-streams package is required to install the IBM Data Cataloging service. If you plan to use the IBM IBM Data Cataloging service, retain it in the commands, or else you can skip it.
- The **cloudnative-pg** package is required to install the CAS service. If you plan to use the CAS service, add it in the commands, or else you can skip it.
- The **kubevirt-hyperconverged**, **metallb-operator**, and **multicluster-engine** operators are required to deploy Hosted Control Plane clusters.
- The **kubernetes-nmstate-operator** package is required for Stage2, that manages node network configuration
- The **node-feature-discovery(nfd)** package is required when the cluster contains GPU nodes or specialized hardware that needs to be discovered and labeled. Essential for GPU enabled workloads and hardware accelerated services.
- The **rhods-operator** package is required for Software Hub and IBM watsonx services.
- The **openshift-cert-manager-operator** package is required for Software Hub and IBM watsonx services to manage TLS certificates and secure communications between service components.

4. Run the following **oc** command to mirror the images from the specified image set configuration to a specified registry:

```
oc mirror --v2 --config imageset-config-redhatoperator.yaml --workspace file://./ docker://${TARGET_PATH} --dest-tls-verify=false
```

This can take 1-2 hours to complete.

Example output:

```
Rendering catalog image "<TARGET_PATH>/redhat/redhat-operator-index:v4.16" with file-based catalog
Writing image mapping to oc-mirror-workspace/results-1693810862/mapping.txt
Writing CatalogSource manifests to oc-mirror-workspace/results-1693810862
Writing ICSP manifests to oc-mirror-workspace/results-1693810862
```

5. Important: You can run steps 5, 6, and 7 only when the cluster is up for a freshly installed rack.
Apply the generated **ImageDigestMirrorSet** to the cluster:

```
oc apply -f working-dir/cluster-resources/idms-oc-mirror.yaml
```

Example **ImageDigestMirrorSet**:

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  annotations:
    createdAt: Tuesday, 07-Apr-26 14:48:07 UTC createdBy: oc-mirror v2 oc-mirror_version: 4.20.0-202511252120.p2.g5b23c4a.asse
```

```
  name: idms-operator-0
spec:
  imageDigestMirrors:
  - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/migration-toolkit-virtualization
    source: registry.redhat.io/migration-toolkit-virtualization
  - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/openshift4
    source: registry.redhat.io/openshift4
  - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/amq-streams
    source: registry.redhat.io/amq-streams
  - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/rhel9
    source: registry.redhat.io/rhel9
  - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/workload-availability
    source: registry.redhat.io/workload-availability
  - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/cert-manager
    source: registry.redhat.io/cert-manager
  - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/rhoai
    source: registry.redhat.io/rhoai
  - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/multicluster-engine
    source: registry.redhat.io/multicluster-engine
  - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/oadp
    source: registry.redhat.io/oadp
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/rhel7
    source: registry.redhat.io/rhel7
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new
    source: registry.redhat.io
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/container-native-virtualization
    source: registry.redhat.io/container-native-virtualization
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/rhaii-early-access
    source: registry.redhat.io/rhaii-early-access
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/lvms4
    source: registry.redhat.io/lvms4
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/rhaiis
    source: registry.redhat.io/rhaiis
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/rhelai1
    source: registry.redhat.io/rhelai1
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/ubi9
    source: registry.redhat.io/ubi9
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/ubi8
    source: registry.redhat.io/ubi8
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/cluster-observability-operator
    source: registry.redhat.io/cluster-observability-operator
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/openshift-service-mesh
    source: registry.redhat.io/openshift-service-mesh
   - mirrors:
    - r11-ru22-w-l.quay-service.fusion.tadn.ibm.com:8443/temptest/test-rh-new/rhel8
    source: registry.redhat.io/rhel8
```

6. Apply the generated `CatalogSource` to the cluster:

```
oc apply -f catalogSource-cs-redhat-operator-index.yaml
oc apply -f catalogSource-cs-certified-operator-index.yaml
```

For example:

```
oc apply -f working-dir/cluster-resources/cs-redhat-operator-index-v4-20.yaml
```

7. For offline environments, update the `redhat-operators` CatalogSource image with the following mirrored image:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: redhat-operators
  namespace: openshift-marketplace
spec:
  displayName: Red Hat Operators
  image: $TARGET_PATH/redhat/redhat-operator-index:v$OCP_VERSION
  publisher: Red Hat
  sourceType: grpc
```

Mirroring individual images

There may be scenarios wherein you may want to mirror individual images separately.

You may choose to mirror individual images to suit customized requirements, manage resource usage, or greater control over the mirroring.

Mirroring IBM Fusion images

Mirror the IBM Fusion images to your enterprise registry.

Before you begin

- Before you mirror, go through the prerequisites. See Mirroring prerequisites.
- Use either podman or docker commands to login to the registries. The examples in the procedure used docker; if you are using podman, substitute `podman` for `docker` in the commands.

About this task

Use this procedure only necessary if you plan to mirror specific sub-components of IBM Fusion instead of following the full end-to-end mirroring instructions. For more information about the end-to-end mirroring of IBM Fusion and all its services, see End-to-end mirroring of IBM Fusion and its services images to the enterprise registry.

High level tasks for mirroring:

- Log in to the relevant Docker registries.
- Define some convenience variables that are used to configure and mirror the Fusion base component.
- Run the `oc-mirror` CLI tool that mirrors the Fusion base components.
- Run the mirroring steps for the additional components you want to include in your installation.

Note: If you choose to mirror specific sub-components, it is recommended that you use the same `TARGET_PATH` variable so that all the components mirror in the same location, and the generated `ImageDigestMirrorSet` is same across all components.

Procedure

1. Run the following command to login to Docker registry with your Red Hat® enterprise credentials:
   Note: If you do not have Docker, you can use Podman.

   ```
   docker login registry.redhat.io -u <Red Hat enterprise registry username> -p <Red Hat enterprise registry password>
   ```

   Log in to the IBM Entitled Container Registry using the IBM entitlement key:

   ```
   docker login cp.icr.io -u cp -p <your entitlement key>
   ```

   Note: Ensure that your entitlement key for IBM Fusion contains the correct entitlement.
   Set the following environment variables:
   Important: Ensure you review the offline upgrade considerations before proceeding with upgrade mirroring.

   ```
   export LOCAL_SECRET_JSON="<relative path to pull-secret.json>"
   export LOCAL_ISF_REGISTRY="<Your container registry host>:<port>"
   LOCAL_ISF_REPOSITORY="<Your image path>"
   TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
   export CASE_NAME=ibm-spectrum-fusion-sds
   export CASE_VERSION=2.13.0
   ```

   Note: Port is a non-mandatory value when setting the `LOCAL_ISF_REGISTRY` variable. You can ignore this if your enterprise registry is accessible and has a secure connection.
   Sample value for without port:

   ```
   export LOCAL_ISF_REGISTRY="registryhost.com"
   ```

   See the following sample values:

   ```
   export LOCAL_ISF_REGISTRY="registryhost.com:443"
   export LOCAL_ISF_REPOSITORY="fusion-mirror"
   ```

   `LOCAL_ISF_REGISTRY` is your entitlement registry.

   `LOCAL_ISF_REPOSITORY` is the image path in which you want to mirror the images. You can choose your own repository paths. For example, sds-images/isf or sds-images or sds-images/isf or sds-images.

2. Run the command to login to the Docker registry with your enterprise registry credentials.

   ```
   docker login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
   ```

3. Create the image set configuration for IBM Fusion.
   `imageset-config-fusion.yaml`

   ```
   cat << EOF > imageset-config-fusion.yaml
   kind: ImageSetConfiguration
   apiVersion: mirror.openshift.io/v2alpha1
   mirror:
     operators:
     - catalog: icr.io/cpopen/isf-operator-software-
   catalog@sha256:569b3f5158af370abe3dbe049c53be55d32f239a717ca74f1f5e544c697e8f21
       packages:
       - name: isf-operator
         channels:
         - name: v2.0
         minVersion: 2.13.0
   ```

```
      maxVersion: 2.13.0
  additionalImages:
  - name: icr.io/cpopen/isf-operator-software-catalog:2.13.0-
13849@sha256:569b3f5158af370abe3dbe049c53be55d32f239a717ca74f1f5e544c697e8f21
EOF
```

4. Run the following oc-mirror command to mirror the images.

```
oc mirror --v2 --config imageset-config-fusion.yaml --workspace file://./ docker://${TARGET_PATH} --dest-tls-verify=false
```

5. Important: You can run steps 5 and 6 only when the cluster is up for a freshly installed rack.
   Apply the IDMS file to your cluster:
   Note: Replace the variable `$TARGET_PATH` with your registry details where images are mirrored.

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: ibm-spectrum-fusion
spec:
  imageDigestMirrors:
  - mirrors:
    - $TARGET_PATH/cp
    source: cp.icr.io/cp
  - mirrors:
    - $TARGET_PATH/cpopen
    source: icr.io/cpopen
  - mirrors:
    - $TARGET_PATH/openshift4
    source: registry.redhat.io/openshift4
```

6. Create a `CatalogSource`.
   Note: Replace the variable `$TARGET_PATH` with your registry details where images are mirrored.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: isf-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-spectrum-fusion-sds-2.13.0
  publisher: IBM
  image: ${TARGET_PATH}/cpopen/isf-operator-software-
catalog@sha256:569b3f5158af370abe3dbe049c53be55d32f239a717ca74f1f5e544c697e8f21
  sourceType: grpc
```

7. Follow the steps to mirror the Usage Metering Service (UMS):
   Important: This step is required for mirroring IBM Fusion images. Do not skip.
   a. Create the image set configuration for UMS.

```
cat imageset-config-ums.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v2alpha1
mirror:
 operators:
 - catalog: icr.io/cpopen/ibm-usage-metering-operator-
catalog@sha256:a97cef95c73ac6dacea010818757b4ed9b7440f4c0ec4154a091f248c68d1c9e
   full: true
   packages:
    - name: ibm-usage-metering-operator
      channels:
        - name: v1.0
 additionalImages:
 - name: icr.io/cpopen/ibm-usage-metering-operator-catalog:2.13.0-
24525@sha256:a97cef95c73ac6dacea010818757b4ed9b7440f4c0ec4154a091f248c68d1c9e
EOF
```

   b. Run the following oc command to mirror the images.

```
oc mirror --v2 --config imageset-config-ums.yaml --workspace file://./ docker://${TARGET_PATH} --dest-tls-
verify=false
```

   c. Important: You can run this step only when the cluster is up for a freshly installed rack.
      Apply the `ImageDigestMirrorSet` file to your cluster:
      Note: Replace the variable `$TARGET_PATH` with your registry details where images are mirrored.

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: ibm-usage-metering
spec:
  imageDigestMirrors:
  - mirrors:
    - ${TARGET_PATH}/cpopen
    source: icr.io/cpopen
```

      In case of any failure, refer Troubleshooting and known issues in offline mirroring in offline mirroring to get the resolution steps.

Mirroring IBM Storage Scale images

Mirror the IBM Storage Scale images to your enterprise registry.

Before you begin

- Use this procedure to mirror IBM Storage Scale images to enable the Global Data Platform service in IBM Fusion on Red Hat® OpenShift® Container Platform 4.20 or earlier.
- For Red Hat OpenShift Container Platform 4.21:

  IBM Storage Scale Container Native Storage Access (CNSA) is integrated with Fusion Data Foundation through the remote mount feature.

  To enable the Global Data Platform service in IBM Fusion:
  1. Mirror the Fusion Data Foundation images as described in Mirroring Fusion Data Foundation images.
  2. Mirror the IBM Storage Scale 6.0.1.0 images as described in Mirroring IBM Storage Scale 6.0.1.0 images.
- Before you mirror, go through the prerequisites. See Mirroring prerequisites.
- Use either podman or docker commands to login to the registries. The examples in the procedure used docker; if you are using podman, substitute `podman` for `docker` in the commands.
- For more information on offline setup for network restricted Red Hat OpenShift Container Platform clusters, see Air-gaped installs.

Procedure

1. Log in to the IBM Entitled Container Registry using the IBM entitlement key.

   ```
   docker login cp.icr.io -u cp -p <your entitlement key>
   ```

   Note: Ensure that your entitlement key for IBM Fusion contains the correct entitlement.
   Set the following environment variables:

   ```
   export LOCAL_ISF_REGISTRY="<Your enterprise registry host>:<port>"
   export LOCAL_ISF_REPOSITORY="<Your image path>"
   export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
   #verify target repository value echo "$TARGET_PATH"
   ```

   Note: Port is a non-mandatory value when setting the `LOCAL_ISF_REGISTRY` variable. You can ignore this if your enterprise registry is accessible and has a secure connection.
   Sample value for without port:

   ```
   export LOCAL_ISF_REGISTRY="registryhost.com"
   ```

   See the following sample values:

   ```
   export LOCAL_ISF_REGISTRY="registryhost.com:443"
   export LOCAL_ISF_REPOSITORY="fusion-mirror"
   ```

   `LOCAL_ISF_REGISTRY` is your entitlement registry with port, if you are not using the default port 443.

   `LOCAL_ISF_REPOSITORY` is the image path in which you want to mirror the images. You can choose your own repository paths. For example, sds-images/isf or sds-images.

2. Run the command to login to the Docker registry with your enterprise registry credentials.

   ```
   docker login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
   ```

3. From the mirroring host, run the following copy command to copy IBM Storage Scale images to the host:
   Note: Ensure that you are logged into the source and destination repositories via docker login command.

   ```
   skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/data-management/ibm-spectrum-scale-daemon@sha256:feac3019cd5384ac327b7cf775177e146da024ecac2315d4312c8bb12bc3e417 docker://$TARGET_PATH/data-management/ibm-spectrum-scale-daemon@sha256:feac3019cd5384ac327b7cf775177e146da024ecac2315d4312c8bb12bc3e417
   skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/ibm-spectrum-scale-core-init@sha256:a362af4bdbbf8a1f867df22646439cbd791ef1038310c4a81ed81128f0cef70d docker://$TARGET_PATH/ibm-spectrum-scale-core-init@sha256:a362af4bdbbf8a1f867df22646439cbd791ef1038310c4a81ed81128f0cef70d
   skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/ibm-spectrum-scale-gui@sha256:05a1bb16e7ca717b279935e9678a92eaec97b530878f307418e95171d9934d2b docker://$TARGET_PATH/ibm-spectrum-scale-gui@sha256:05a1bb16e7ca717b279935e9678a92eaec97b530878f307418e95171d9934d2b
   skopeo copy --all --preserve-digests
   docker://cp.icr.io/cp/gpfs/postgres@sha256:ef257d85f76e48da1c64832459b59fcaba1a4dac97bf5d7450c77753542eee94
   docker://$TARGET_PATH/postgres@sha256:ef257d85f76e48da1c64832459b59fcaba1a4dac97bf5d7450c77753542eee94
   skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/ibm-spectrum-scale-logs@sha256:c0d8156a277041e4c8eb7206c428a8c8a1728b5117dd3c91d8bfe6672349b42b docker://$TARGET_PATH/ibm-spectrum-scale-logs@sha256:c0d8156a277041e4c8eb7206c428a8c8a1728b5117dd3c91d8bfe6672349b42b
   skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/ibm-spectrum-scale-pmcollector@sha256:61163d62fc93fef53bfbead12d0bbc32eebb612fa0dc3afe8e938bca318aac25 docker://$TARGET_PATH/ibm-spectrum-scale-pmcollector@sha256:61163d62fc93fef53bfbead12d0bbc32eebb612fa0dc3afe8e938bca318aac25
   skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/ibm-spectrum-scale-monitor@sha256:856b7b4978382ecacfa8235cc80bc87e721f768ef6999c23f3e4a0db66c589f2 docker://$TARGET_PATH/ibm-spectrum-scale-monitor@sha256:856b7b4978382ecacfa8235cc80bc87e721f768ef6999c23f3e4a0db66c589f2
   skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/ibm-spectrum-scale-grafana-bridge@sha256:d423ca26549177b434efddd1fb21932f31110c288cfd77a2390d3bdf8af8876d docker://$TARGET_PATH/ibm-spectrum-scale-grafana-bridge@sha256:d423ca26549177b434efddd1fb21932f31110c288cfd77a2390d3bdf8af8876d
   skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/ibm-spectrum-scale-coredns@sha256:d5f4690898d9468a22d92f6da530cd28966dffc4e8519de89236954a3815dd02 docker://$TARGET_PATH/ibm-spectrum-scale-coredns@sha256:d5f4690898d9468a22d92f6da530cd28966dffc4e8519de89236954a3815dd02
   skopeo copy --all --preserve-digests docker://icr.io/cpopen/ibm-spectrum-scale-must-gather@sha256:a9c18c3f77a8ae89647b238b6504422fabc5bdc0aa59f45a7e179e2472086ec4 docker://$TARGET_PATH/ibm-spectrum-scale-must-gather@sha256:a9c18c3f77a8ae89647b238b6504422fabc5bdc0aa59f45a7e179e2472086ec4
   skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/csi/csi-
   ```

```
snapshotter@sha256:bc7be893ecc3ad524194aa6573b2f5c06cd469bdf21a500ab6c99c2ba1c4d64d docker://$TARGET_PATH/csi/csi-
snapshotter@sha256:bc7be893ecc3ad524194aa6573b2f5c06cd469bdf21a500ab6c99c2ba1c4d64d
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/csi/csi-
attacher@sha256:5aaefc24f315b182233c8b6146077f8c32e274d864cb03c632206e78bd0302da docker://$TARGET_PATH/csi/csi-
attacher@sha256:5aaefc24f315b182233c8b6146077f8c32e274d864cb03c632206e78bd0302da
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/csi/csi-
provisioner@sha256:bb057f866177d5f4139a1527e594499cbe0feeb67b63aaca8679dfdf0a6016f9 docker://$TARGET_PATH/csi/csi-
provisioner@sha256:bb057f866177d5f4139a1527e594499cbe0feeb67b63aaca8679dfdf0a6016f9
skopeo copy --all --preserve-digests
docker://cp.icr.io/cp/gpfs/csi/livenessprobe@sha256:88092d100909918ae0a768956cf78c88bc59cd7232720f7cdbdfb5d2e235001e
docker://$TARGET_PATH/csi/livenessprobe@sha256:88092d100909918ae0a768956cf78c88bc59cd7232720f7cdbdfb5d2e235001e
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/csi/csi-node-driver-
registrar@sha256:5244abbe87e01b35adeb8bb13882a74785df0c0619f8325c9e950395c3f72a97 docker://$TARGET_PATH/csi/csi-node-
driver-registrar@sha256:5244abbe87e01b35adeb8bb13882a74785df0c0619f8325c9e950395c3f72a97
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/csi/csi-
resizer@sha256:5e7cbb63fd497fa913caa21fee1a69f727c220c6fa83c5f8bb0995e2ad73a474 docker://$TARGET_PATH/csi/csi-
resizer@sha256:5e7cbb63fd497fa913caa21fee1a69f727c220c6fa83c5f8bb0995e2ad73a474
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/gpfs/csi/ibm-spectrum-scale-csi-
driver@sha256:dd589aa2abd408d2e9c8f6a0b001db27802f1c3175a95c54f7c2d4993f6ae8d4 docker://$TARGET_PATH/csi/ibm-spectrum-
scale-csi-driver@sha256:dd589aa2abd408d2e9c8f6a0b001db27802f1c3175a95c54f7c2d4993f6ae8d4
skopeo copy --all --preserve-digests docker://icr.io/cpopen/ibm-spectrum-scale-
operator@sha256:138ecf352bb886d6e2d8c81c6d94bad4b4b16ad36be472f5d858f060c845907a docker://$TARGET_PATH/ibm-spectrum-scale-
operator@sha256:138ecf352bb886d6e2d8c81c6d94bad4b4b16ad36be472f5d858f060c845907a
```

Note: All commands must be successful.

4. In the Red Hat OpenShift Container Platform cluster, add **ImageDigestMirrorSet**.
   Note:
   - Replace the variable **$TARGET_PATH** with your registry details where images are mirrored.
   - Do not add white spaces while you update the mirror path in ImageDigestMirrorSet. If you want to do so, add values within double quotation marks. For example, if the new path where the images are mirrored is **registryhost.com:443/isf-new-path**, then encapsulate within double quotation marks ("") while you edit to avoid white spaces in the line.

```
- mirrors:
    - "registryhost.com:443/isf-new-path"
    - registryhost.com:443/old-path
   source: cp.icr.io/cp/spectrum/scale
```

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: isf-scale-idms
spec:
  imageDigestMirrors:
    - mirrors:
        - $TARGET_PATH
      source: cp.icr.io/cp/spectrum/scale
    - mirrors:
        - $TARGET_PATH
      source: cp.icr.io/cp/gpfs
    - mirrors:
        - $TARGET_PATH
      source: icr.io/cpopen
```

Mirroring IBM Storage Scale 6.0.1.0 images

Mirror IBM Storage Scale 6.0.1.0 images to your enterprise registry for installation in a disconnected environment by using the **oc-mirror** plug-in.

Before you begin

Before you mirror images, review the prerequisites in Mirroring prerequisites.

- Ensure that Red Hat® OpenShift® Container Platform version 4.21 is installed.
- Ensure that the Red Hat operator images are mirrored. For more information, see Mirroring Red Hat operator images to enterprise registry.
- Ensure that the Fusion Data Foundation images are mirrored. For more information, see Mirroring Fusion Data Foundation images.
- Use either the **podman** or **docker** commands to log in to the registries. The examples in this procedure use **docker**. If you use **podman**, replace **docker** with **podman** in the commands.

- For more information about offline setup for network-restricted Red Hat OpenShift Container Platform clusters, see Air-gaped installs.

Procedure

1. Log in to the IBM Entitled Container Registry by using your IBM entitlement key:
   Command example:

```
docker login cp.icr.io -u cp -p <your entitlement key>
```

2. Set the following environment variables:
   Command example:

```
export LOCAL_ISF_REGISTRY="<your enterprise registry host>:<port>"
export LOCAL_ISF_REPOSITORY="<your image path>"
export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
```

3. Log in to your enterprise registry:
   Command example:

```
docker login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
```

4. Create the image set configuration for Fusion Data Foundation.
   Command example where Fusion Data Foundation version is 4.21.7:

```
cat << EOF > image-set-config.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
mirror:
  additionalImages:
  - name: cp.icr.io/cp/gpfs/csi/csi-
attacher:v4.11.0@sha256:b74b05b39501565022883fc128002b4cb857a7bb6c858606bcb3fdedba0b0b80
  - name: cp.icr.io/cp/gpfs/csi/csi-node-driver-
registrar:v2.16.0@sha256:ab482308a4921e28a6df09a16ab99a457e9af9641ff44fb1be1a690d07ce8b70
  - name: cp.icr.io/cp/gpfs/csi/csi-
provisioner:v6.2.0@sha256:6be9f63ca4caa6c46aae55aa372500949d8a21473d72f819da1f746076b32d4e
  - name: cp.icr.io/cp/gpfs/csi/csi-resizer:v2.1.0@sha256:589e525cddef6d768e68da1f0bc9ffd0a24bf3add3dd010648eb7189976fde79
  - name: cp.icr.io/cp/gpfs/csi/csi-
snapshotter:v8.5.0@sha256:da081c27e8a6d91f36042c1942362d0515ced8d06e18c11b8f893e58c4d6d797
  - name: cp.icr.io/cp/gpfs/csi/ibm-spectrum-scale-csi-
driver:v3.1.0@sha256:35c2c45c0a8f6504cf50dda57fd0c827244e822e02febf449a37630fc6d01b9d
  - name:
cp.icr.io/cp/gpfs/csi/livenessprobe:v2.18.0@sha256:c4cc074199c045dd73ab85f28897e2a32f4d6f38ffdba4f3b13b8007ccbd3570
  - name: cp.icr.io/cp/gpfs/data-management/ibm-spectrum-scale-
daemon:v6.0.1.0@sha256:3651a79cfd42e67416995af3442b667beef09c9c1417e406b7be20cb63497ddf
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-core-
init:v6.0.1.0@sha256:1775f4d2c51ae9bef6d0d129c79efcfefb4d6d4008445d57b852d4c4397b2fe2
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
coredns:v6.0.1.0@sha256:781db6ee6019ae2468b57f48abc17033ccfcdb88beb555b2797ee4fe32a2731b
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-grafana-
bridge:v6.0.1.0@sha256:3843a5db15d214355d7c80751b7ab771cbaef9be025eea04745386d9210914e5
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
gui:v6.0.1.0@sha256:fd85ae58cfdf4ef48b300f542967a0b2e626ff4015af86a5f31e648b76d11160
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
logs:v6.0.1.0@sha256:16754a6b3e1eaac73a3df6d6ded01d31e0f5d0cc75dc6f30d1cb8043d2dc0685
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
monitor:v6.0.1.0@sha256:a15797ec26b5e6b54de4396ed4fdb9303169908c04337694ff2d2f2e9372df87
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
pmcollector:v6.0.1.0@sha256:10dd10e7ab9c32d36b31924c266671573c48fe1fd4c43cb1a6d89d7644482a4c
  - name: cp.icr.io/cp/gpfs/postgres:18.1-
alpine3.23@sha256:aa6eb304ddb6dd26df23d05db4e5cb05af8951cda3e0dc57731b771e0ef4ab29
  - name: icr.io/cpopen/ibm-spectrum-scale-must-
gather:v6.0.1.0@sha256:39d5a03dcc657ce704299767e09408e52be040edf11d8a2cf4b72864178e4535
  - name: icr.io/cpopen/ibm-spectrum-scale-operator-
bundle:v6.0.1.0@sha256:b29e753b855f26561dc320c90f70940d96400d79d04f6870a8aef3ca72b6e5f0
  - name: icr.io/cpopen/ibm-spectrum-scale-
operator:v6.0.1.0@sha256:5e2095f878f45b561e17fc00725d6ca69653d12f7693d295b0a53a40046f1a45
EOF
```

5. Run the following command to mirror the images:
   Command example:

```
oc mirror --v2 --config image-set-config.yaml --workspace file://./ docker://${TARGET_PATH} --dest-tls-verify=false
```

6. Create an `ImageDigestMirrorSet` custom resource named `isf-fdf-idms.yaml`.
   Important: Replace `$TARGET_PATH` with the registry path where the images are mirrored.
   Command example:

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: ibm-spectrum-scale-dme-bundle
spec:
  imageDigestMirrors:
  - mirrors:
    - $TARGET_PATH/cp
    source: cp.icr.io/cp
  - mirrors:
    - $TARGET_PATH/cpopen
    source: icr.io/cpopen
```

7. Apply the `ImageDigestMirrorSet` configuration.
   Command example:

```
oc apply -f isf-fdf-idms.yaml
```

8. Verify that the images are pulled from your internal registry and that the operator pod is in the Running state.
   Command example:

```
kubectl get pods -lapp.kubernetes.io/name=operator -n ibm-spectrum-scale
```

   Example output:

```
NAME                                                  READY   STATUS    RESTARTS   AGE
ibm-spectrum-scale-controller-manager-64d548f576-znrgc   1/1     Running   0          37m
```

Mirroring Fusion Data Foundation images

Mirror the Fusion Data Foundation images deployed on Red Hat® OpenShift® Container Platform to your enterprise registry by using `ImageDigestMirrorSet`.

Before you begin

- Use this procedure to mirror Fusion Data Foundation images to enable the Fusion Data Foundation service in IBM Fusion.
- To enable the Global Data Platform service in IBM Fusion on Red Hat OpenShift Container Platform 4.21:

    1. Use this procedure to mirror the Fusion Data Foundation images.
    2. Mirror the IBM Storage Scale 6.0.1.0 images as described in Mirroring IBM Storage Scale 6.0.1.0 images.
- Before you mirror, go through the prerequisites. See Mirroring prerequisites.
- Use either podman or docker commands to login to the registries. The examples in the procedure used docker; if you are using podman, substitute `podman` for `docker` in the commands.
- For more information about the version support, see IBM Fusion support matrix.

About this task

The Fusion Data Foundation 4.16 with Red Hat OpenShift Container Platform 4.16 is used in the procedure as an example. Replace it with your appropriate supported version.

Procedure

1. Run the following command to login to Docker registry with your Red Hat® enterprise credentials:

   ```
   docker login registry.redhat.io -u <Red Hat enterprise registry username> -p <Red Hat enterprise registry password>
   ```

2. Log in to the IBM Entitled Container Registry by using the IBM entitlement key:

   ```
   docker login cp.icr.io -u cp -p <your entitlement key>
   ```

   Note: Ensure that your entitlement key for IBM Fusion contains the correct entitlement.
   Set the following environment variables:

   ```
   export LOCAL_ISF_REGISTRY="<Your enterprise registry host>:<port>"
   export LOCAL_ISF_REPOSITORY="<Your image path>"
   export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
   export OCP_VERSION="<Your ocp version, eg 4.16>"
   ```

   Note: Port is a non-mandatory value when you set the `LOCAL_ISF_REGISTRY` variable. If your enterprise registry is accessible and has a secure connection, then you can ignore it.
   Sample value for without port:

   ```
   export LOCAL_ISF_REGISTRY="registryhost.com"
   ```

   See the following sample values:

   ```
   export LOCAL_ISF_REGISTRY="registryhost.com:443"
   export LOCAL_ISF_REPOSITORY="fusion-mirror"
   ```

   `LOCAL_ISF_REGISTRY` is your entitlement registry.

   `LOCAL_ISF_REPOSITORY` is the image path in which you want to mirror the images. You can choose your own repository paths. For example, in 2.13.0, the name can be imagessds-images/isf or sds-images.

3. Run the command to login to the Docker registry with your enterprise registry credentials:

   ```
   docker login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
   ```

4. Create the image set configuration for Fusion Data Foundation.
   If you are using Fusion Data Foundation 4.21, then use the following image set configuration:

   `imageset-config-fdf.yaml`

   ```
   cat << EOF > imageset-config-fdf.yaml
   kind: ImageSetConfiguration
   apiVersion: mirror.openshift.io/v2alpha1
   mirror:
    operators:
    - catalog: icr.io/cpopen/isf-data-foundation-catalog:v$OCP_VERSION
      packages:
        - name: "cephcsi-operator"
        - name: "mcg-operator"
        - name: "ocs-operator"
        - name: "odf-csi-addons-operator"
        - name: "odf-multicluster-orchestrator"
        - name: "odf-operator"
        - name: "odr-cluster-operator"
        - name: "odr-hub-operator"
        - name: "ocs-client-operator"
        - name: "odf-prometheus-operator"
        - name: "recipe"
        - name: "rook-ceph-operator"
        - name: "odf-dependencies"
        - name: "odf-external-snapshotter-operator"
        - name: "ibm-spectrum-scale-operator"
        - name: "cnsa-dependencies"
   EOF
   ```

   If you are using Fusion Data Foundation 4.20, then use the following image set configuration:

`imageset-config-fdf.yaml`

```
cat << EOF > imageset-config-fdf.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v2alpha1
mirror:
 operators:
 - catalog: icr.io/cpopen/isf-data-foundation-catalog:v$OCP_VERSION
   packages:
     - name: "cephcsi-operator"
     - name: "mcg-operator"
     - name: "ocs-operator"
     - name: "odf-csi-addons-operator"
     - name: "odf-multicluster-orchestrator"
     - name: "odf-operator"
     - name: "odr-cluster-operator"
     - name: "odr-hub-operator"
     - name: "ocs-client-operator"
     - name: "odf-prometheus-operator"
     - name: "recipe"
     - name: "rook-ceph-operator"
     - name: "odf-dependencies"
     - name: "odf-external-snapshotter-operator"
EOF
```

If you are using Fusion Data Foundation 4.18 and 4.19, then use the following image set configuration:

`imageset-config-fdf.yaml`

```
cat << EOF > imageset-config-fdf.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v2alpha1
mirror:
 operators:
 - catalog: icr.io/cpopen/isf-data-foundation-catalog:v$OCP_VERSION
   packages:
     - name: "cephcsi-operator"
     - name: "mcg-operator"
     - name: "ocs-operator"
     - name: "odf-csi-addons-operator"
     - name: "odf-multicluster-orchestrator"
     - name: "odf-operator"
     - name: "odr-cluster-operator"
     - name: "odr-hub-operator"
     - name: "ocs-client-operator"
     - name: "odf-prometheus-operator"
     - name: "recipe"
     - name: "rook-ceph-operator"
     - name: "odf-dependencies"
EOF
```

5. Run the following oc command to mirror the images.

   ```
   oc mirror --v2 --config imageset-config-fdf.yaml --workspace file://./ docker://${TARGET_PATH} --dest-tls-verify=false
   ```

6. Run the following command to check whether the image got successfully mirrored to the registry:
   Note: If you are using a self signed registry, then export the `SSL_CERT_DIR=/root/quay-install/quay-rootCA` before running this command.
   The `/root/quay-install/quay-rootCA` path is where certificates are placed. For more information, see Red Hat solution.

   ```
   oc mirror list operators --catalog $TARGET_PATH/cpopen/isf-data-foundation-catalog:v$OCP_VERSION
   ```

   Sample output:

   ```
   WARN[0033] DEPRECATION NOTICE:

   Sqlite-based catalogs and their related subcommands are deprecated. Support for them will be removed in a future release. Plea
   to the new file-based catalog format.

   NAME                                DISPLAY NAME                DEFAULT                 CHANNEL

   mcg-operator                        NooBaa                      Operator                stable-4.16

   ocs-client-operator                 OpenShift Data Foundation   Client                  stable-4.16

   ocs-operator                        Container                   Storage                 stable-4.16

   odf-csi-addons-operator             CSI                         Addons                  stable-4.16

   odf-multicluster-orchestrator       Multicluster                Orchestrator            stable-4.16

   odf-operator                        IBM Storage Fusion          Data Foundation         stable-4.16

   odr-cluster-operator                DR Cluster                  Operator                stable-4.16

   odr-hub-operator                    DR Hub                      Operator                 stable-4.16
   ```

7. Ensure that the local-storage-operator and lvms-operator operator packages are present in your cluster.
   Note: If you have not mirrored local-storage-operator and lvms-operator from the Red Hat packages previously, then follow the steps that are provided in the Mirroring Red Hat operator images to enterprise registry.
8. Note: You can run steps 8, 9, and 10 only when the cluster is up for a freshly installed rack.
   Create an `ImageDigestMirrorSet` custom resource for Fusion Data Foundation. Apply the configuration by running the command:

   ```
   oc apply -f isf-fdf-idms.yaml
   ```

   See the following `ImageDigestMirrorSet`:

Note: Replace the variable **$TARGET_PATH** with your registry details where images are mirrored.

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  labels:
    operators.openshift.org/catalog: "true"
  name: isf-fdf-idms
spec:
  imageDigestMirrors:
  - mirrors:
    - $TARGET_PATH/openshift4
    source: registry.redhat.io/openshift4
  - mirrors:
    - $TARGET_PATH/redhat
    source: registry.redhat.io/redhat
  - mirrors:
    - $TARGET_PATH/rhel9
    source: registry.redhat.io/rhel9
  - mirrors:
    - $TARGET_PATH/rhel8
    source: registry.redhat.io/rhel8
  - mirrors:
    - $TARGET_PATH/cp/df
    source: cp.icr.io/cp/df
  - mirrors:
    - $TARGET_PATH/cpopen
    source: cp.icr.io/cpopen
  - mirrors:
    - $TARGET_PATH/cpopen
    source: icr.io/cpopen
  - mirrors:
    - $TARGET_PATH/cp/df/ceph
    source: cp.icr.io/cp/df/ceph
  - mirrors:
    - $TARGET_PATH/odf4
    source: registry.redhat.io/odf4
  - mirrors:
    - $TARGET_PATH/lvms4
    source: registry.redhat.io/lvms4
  - mirrors:
    - $TARGET_PATH/rhceph
    source: registry.redhat.io/rhceph
```

9. Create a **CatalogSource** named **redhat-operators**. Apply the **CatalogSource** by running the command:

   ```
   oc apply -f redhat-operators-catalogsource.yaml
   ```

   Note: Do this step when the Fusion Data Foundation uses the local disk.

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: redhat-operators
     namespace: openshift-marketplace
   spec:
     displayName: Red Hat Operators
     image: $TARGET_PATH/redhat/redhat-operator-index:v$OCP_VERSION
     publisher: Red Hat
     sourceType: grpc
   ```

10. Create an **ImageTagMirrorSet** for Fusion Data Foundation. Apply the configuration by running the command:

    ```
    oc apply -f isf-fdf-imagetagmirrorset.yaml
    ```

    See the following **ImageDigestMirrorSet**:

    ```
    apiVersion: config.openshift.io/v1
    kind: ImageTagMirrorSet
    metadata:
      name: isf-fdf
    spec:
      imageTagMirrors:
        - mirrors:
            - $TARGET_PATH/cpopen/isf-data-foundation-catalog
          source: icr.io/cpopen/isf-data-foundation-catalog
          mirrorSourcePolicy: AllowContactingSource
    ```

What to do next

To enable the Global Data Platform service in IBM Fusion on Red Hat OpenShift Container Platform 4.21, mirror the IBM Storage Scale 6.0.1.0 images as described in Mirroring IBM Storage Scale 6.0.1.0 images.

Mirroring Backup & Restore images

Mirror the Backup & Restore images to your enterprise registry.

Before you begin

1. Before you mirror, go through the prerequisites. See Mirroring prerequisites.
2. Use either podman or docker commands to login to the registries. The examples in the procedure used docker; if you are using podman, substitute `podman` for `docker` in the commands.
3. For more information about the version support, see IBM Fusion support matrix.

About this task

High level steps to mirror Backup & Restore service:

1. Log in to the relevant Docker registries
2. Define variables to configure and mirror the Backup & Restore component of IBM Fusion.
3. Run the `oc-mirror` CLI tool to mirror the Backup & Restore components.

Note:

- Use this procedure only necessary if you plan to mirror specific sub-components of IBM Fusion instead of following the full end-to-end mirroring instructions. For more information about the end-to-end mirroring of IBM Fusion and all its services, see End-to-end mirroring of IBM Fusion and its services images to the enterprise registry.
- If you choose to mirror specific sub-components, use the same TARGET_PATH variable so that all the components mirror into the same location, and the generated `ImageDigestMirrorSet` remains the same across all components.
- For more information about Air gap setup for network restricted Red Hat® OpenShift® Container Platform clusters, see Offline setup for network restricted Red Hat OpenShift Container Platform clusters.

Procedure

1. Log in to your Docker registry:
   a. Run the following command to login to the Docker registry with your Red Hat enterprise credentials:

      ```
      docker login registry.redhat.io -u <Red Hat enterprise registry username> -p <Red Hat enterprise registry password>
      ```

   b. To authenticate to quay.io by using the username and password from the pull-secret, do the following steps:
      i. Download the pull-secret, see https://console.redhat.com/openshift/install/pull-secret and follow the instructions.
      See the following sample values:

      ```
      "auths": {
        "quay.io": {
          "auth": "b3BlbnNoaWZ0LXJTczOGY2YzNjNDM2ZWI0JRSDdUQU45RFBWVUNXM0VZQUlVVDBOQTU3VFM2RE1JMg==",
          "email": <email-id>
        }
      }
      }
      ```

      ii. Get the username/password:

      ```
      echo <auth-content> | base64 -decode
      ```

      Example echo command:

      ```
      echo "b3BlbnNoaWZ0LXJTczOGY2YzNjNDM2ZWI0JRSDdUQU45RFBWVUNXM0VZQUlVVDBOQTU3VFM2RE1JMg==" | base64 -decode
      ```

      Example output:

      ```
      openshift-release+ocmaccess0ab5738f6c3c42:CXKR2TSBQH7TAN9
      ```

      Here, `openshift-release+ocmaccess0ab5738f6c3c42` is the user name and `CXKR2TSBQH7TAN9` is the password.
   c. Log in to the IBM Entitled Container Registry using the IBM entitlement key.

      ```
      docker login cp.icr.io -u cp -p <your entitlement key>
      ```

      Note: Ensure that your entitlement key for IBM Fusion contains the correct entitlement.
   d. Set the following environment variables:
      Important: Ensure you review the offline upgrade considerations before proceeding with upgrade mirroring.

      ```
      export LOCAL_ISF_REGISTRY="<Your enterprise registry host>:<port>"
      export LOCAL_ISF_REPOSITORY="<Your image path>"
      export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
      echo "$TARGET_PATH"
      ```

      Note: Port is a non-mandatory value when setting the `LOCAL_ISF_REGISTRY` variable. You can ignore this if your enterprise registry is accessible and has a secure connection.
      Sample value without port:

      ```
      export LOCAL_ISF_REGISTRY="registryhost.com"
      ```

      ```
      export LOCAL_ISF_REGISTRY="registryhost.com:443"
      export LOCAL_ISF_REPOSITORY="fusion-mirror"
      ```

   e. Run the command to login to the Docker registry with your enterprise registry credentials.

      ```
      docker login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
      ```

      `LOCAL_ISF_REGISTRY` is your entitlement registry.

      `LOCAL_ISF_REPOSITORY` is the image path in which you want to mirror the images. You can choose your own repository paths. For example, sds-images/isf or sds-images.

2. Ensure that the `redhat-oadp-operator` and `amq-streams` operator packages are present in your cluster.
   Note: If you have not mirrored `redhat-oadp-operator` and `amq-streams` from the Red Hat packages previously, then follow the steps that are provided in the Mirroring Red Hat operator images to enterprise registry.
   Note:
   - Run the following command to get list of mirrored Red Hat packages available on your cluster.

     ```
     oc get packagemanifests | grep -i "Red Hat Operators"
     ```

   - If you do not get `redhat-oadp-operator` and `amq-streams` operator packages from the previous step, then you must follow the step 2.
   - Ensure that you also add existing packages along with new one in ImageSetConfiguration file. Otherwise, old packages can be lost from the Red Hat operator index image.
3. Run the command to login to the Docker registry with your enterprise registry credentials.
   For docker users:

   ```
   docker login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
   ```

   For podman users:

   ```
   podman login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
   ```

4. Create the image set configuration for Backup & Restore.

   ```
   cat << EOF > imageset-config-bnr.yaml
   kind: ImageSetConfiguration
   apiVersion: mirror.openshift.io/v2alpha1
   mirror:
     operators:
     - catalog: icr.io/cpopen/idp-server-operator-
   catalog@sha256:1b93488e0bf5c421dace9bae65fd992695c1e63fa12e7b82f03cf852d3fa02ad
       packages:
       - name: guardian-dm-operator
         channels:
         - name: v2.0
       - name: guardian-dp-operator
         channels:
         - name: v2.0
       - name: guardian-mongo-operator
         channels:
         - name: v2.0
       - name: ibm-dataprotectionagent
         channels:
         - name: v2.0
       - name: ibm-dataprotectionserver
         channels:
         - name: v2.0
       - name: redis-operator
         channels:
         - name: v2.0
       full: true
     additionalImages:
     - name: icr.io/cpopen/idp-server-operator-catalog:2.13.0-
   26195@sha256:1b93488e0bf5c421dace9bae65fd992695c1e63fa12e7b82f03cf852d3fa02ad
   EOF
   ```

5. Run the following `oc-mirror` command to mirror the images.

   ```
   oc mirror --v2 --config imageset-config-bnr.yaml --workspace file://./ docker://${TARGET_PATH} --dest-tls-verify=false
   ```

6. Important: You can run step 5 only when the cluster is up for a freshly installed rack.
   Apply the `ImageDigestMirrorSet` file to your cluster:
   Note: Replace the variable `$TARGET_PATH` with your registry details where images are mirrored.

   ```
   apiVersion: config.openshift.io/v1
   kind: ImageDigestMirrorSet
   metadata:
     name: ibm-fusion-bnr
   spec:
     imageDigestMirrors:
     - mirrors:
       - ${TARGET_PATH}/cp
       source: cp.icr.io/cp
     - mirrors:
       - ${TARGET_PATH}/cpopen
       source: icr.io/cpopen
     - mirrors:
       - ${TARGET_PATH}/minio
       source: quay.io/minio
     - mirrors:
       - ${TARGET_PATH}/openshift4
       source: registry.redhat.io/openshift4
   ```

   In case of any failure, refer Troubleshooting and known issues in offline mirroring to get the resolution steps.

Mirroring IBM Data Cataloging images

Mirror the IBM Data Cataloging images to your enterprise registry.

Before you begin

- Before you mirror, go through the prerequisites. See Mirroring prerequisites.
- Use either podman or docker commands to login to the registries. The examples in the procedure used docker; if you are using podman, substitute `podman` for `docker` in the commands.
- For more information about the version support, see IBM Fusion support matrix.
- Ensure that you install the `oc-mirror OpenShift` CLI plug-in. For more information about `oc-mirror OpenShift` CLI plug-in installation, see Installing the oc-mirror OpenShift CLI plugin.

About this task

For more information about Air gap setup for network restricted Red Hat® OpenShift® Container Platform clusters, see Offline setup for network restricted Red Hat OpenShift Container Platform clusters.
Note:

- This procedure is only necessary if you plan to enable the IBM Data Cataloging IBM Fusion HCI service.
- The destination registry where all IBM Data Cataloging images are going to be copied need to support both OCI (`application/vnd.oci.image.manifest.v1+json`) and Docker V2 (`application/vnd.docker.distribution.manifest.v2+json`) manifest types.

Use this procedure only necessary if you plan to mirror specific sub-components of IBM Fusion instead of following the full end-to-end mirroring instructions. For more information about the end-to-end mirroring of IBM Fusion and all its services, see End-to-end mirroring of IBM Fusion and its services images to the enterprise registry.

High level tasks of this procedure:

1. Log in to the relevant Docker registries.
2. Define some convenience variables used to configure and mirror the Data Cataloging component of Fusion.
3. Run the `oc-mirror` CLI tool mirrors the Data Cataloging components.

Note: If you choose to mirror specific sub-components, it is recommended that you use the same `TARGET_PATH` variable so that all the components mirror in the same location, and the generated `ImageDigestMirrorSet` is same across all components.

Procedure

1. Ensure that the `amq-streams` operator package is present in your cluster.
   Note: If you have not mirrored `amq-streams` from the Red Hat packages previously, then follow the steps that are provided in the Mirroring Red Hat operator images to enterprise registry.
   Note:
   - Run the following command to get list of mirrored Red Hat packages available on your cluster.

     ```
     oc get packagemanifests | grep -i "Red Hat Operators"
     ```

   - If you do not get `redhat-oadp-operator` operator package from the previous step, then you must follow the step 1.
   - Ensure that you also add existing packages along with new one in ImageSetConfiguration file. Otherwise, old packages can be lost from the Red Hat operator index image.
2. Run the following command to login the Red Hat enterprise registry and the IBM Entitled Container Registry using your credentials.
   Note: Ensure that your entitlement key for IBM Fusion contains the correct entitlement.
   For docker users:

   ```
   docker login registry.redhat.io -u <Red Hat enterprise registry username> -p <Red Hat enterprise registry password>

   docker login cp.icr.io -u cp -p <your entitlement key>
   ```

   For podman users:

   ```
   podman login registry.redhat.io -u <Red Hat enterprise registry username> -p <Red Hat enterprise registry password>

   podman login cp.icr.io -u cp -p <Your entitlement key>
   ```

3. Set the following environment variable.
   Important: Ensure you review the offline upgrade considerations before proceeding with upgrade mirroring.

   ```
   export LOCAL_ISF_REGISTRY="<Your enterprise registry host>:<port>"
   export LOCAL_ISF_REPOSITORY="<Your image path>"
   export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
   ```

   `LOCAL_ISF_REGISTRY` is your entitlement registry.

   `LOCAL_ISF_REGISTRY` is the registry image path in which you want to mirror the images. You can choose your own repository paths. For example, `hci-images/isf` or `hci-images`.

4. Run the command to login to the Docker registry with your enterprise registry credentials.
   For docker users:

   ```
   docker login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
   ```

   For podman users:

   ```
   podman login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
   ```

5. Create the image set configuration for IBM Data Cataloging.

   ```
   cat << EOF > imageset-config-dcs.yaml
   kind: ImageSetConfiguration
   apiVersion: mirror.openshift.io/v2alpha1
   mirror:
     additionalImages:
     - name: cp.icr.io/cp/dcs/auth@sha256:6cb6e582d02fb017dbbc1c736f3409a08823fef56381747f13952bd8a613cabb
     - name: cp.icr.io/cp/dcs/connmgr-scheduler@sha256:cca24250b9bbfdfe8c6c85c613c5fa1a56d2b0fbbb0e48a4c56fc3f50ea338a2
     - name: cp.icr.io/cp/dcs/connmgr@sha256:cbefb44b2d448b36f90e4fca7973ab66bc1fbfd799eaa1c13b5185bf116f6f09
   ```

```
  - name: cp.icr.io/cp/dcs/db-schema@sha256:534c98417f909a248698281bc244ef1441b6a01dccf5a3055242c13956576c9a
  - name: cp.icr.io/cp/dcs/db2whrest@sha256:3233681ab2ec32515653f5659d6c02f8e12fbfddaa9d65747dfe07a62de1bfd2
  - name: cp.icr.io/cp/dcs/dcs-ai-mcp-server@sha256:d8af262ff80ee897797d322bbd8f8382cf4b823033901aea37d5531b27565365
  - name: cp.icr.io/cp/dcs/import-service@sha256:1003a8694eff91c163a72b88c19baeb5cd47a97d152ab64ac910b8b8b693f096
  - name: cp.icr.io/cp/dcs/import-tags-app@sha256:c37128ef0ad7f7add93c72c42c8c1902f75a2907e683af5d7af99d04ff4ea381
  - name: cp.icr.io/cp/dcs/isd-proxy@sha256:5056d6a265b2a37123806976c56722b86c5c20dad6d2f5933838c8d076b1e120
  - name: cp.icr.io/cp/dcs/keystone@sha256:b3128687e35f78bc25cefbe7254f625b5f01f344ae190d161cdbf79057109aa6
  - name: cp.icr.io/cp/dcs/llm-proxy@sha256:fce3aebc677eba45434f16039d3c678ccf4c1c71d709ab3b13b5aebd79c928b6
  - name: cp.icr.io/cp/dcs/metadata-extender@sha256:9a988643df596a1bbfb1094874273c3c1ca879e8a645677090409b4de1f37ac9
  - name: cp.icr.io/cp/dcs/mo-agent-extract@sha256:2f4d7b0d5ada12d39932f687cea031d88beb9cdac791b48cdca29ed23a9860ca
  - name: cp.icr.io/cp/dcs/mo-ubi-init@sha256:927bb277f576f2a772faf8dc9978a6592d5d08bfad3cdce81da180a9095a34c2
  - name: cp.icr.io/cp/dcs/moconsumer@sha256:2f1d3fdec099951cfc5f6788d4a4d944d66ec0aa8e39c9688239f1ecad38729f
  - name: cp.icr.io/cp/dcs/moproducer@sha256:c984c05bc47939e22cec6f4c8c0a9f365b71b7582c3711a36a59b96cc546da78
  - name: cp.icr.io/cp/dcs/must-gather@sha256:3f655e31efb04862c52333f113322a9169ea54d88f627177707453fd48acec50
  - name: cp.icr.io/cp/dcs/policyengine@sha256:ffe2fac7ee8bd49be48a142818ca760c6482018d9a79941296ef8d3424c1949c
  - name: cp.icr.io/cp/dcs/remote-content-search@sha256:7c0a39bcf04c1063fffc866ec8513066beb398dfd28cdb436b21cf7c4754f7c8
  - name: cp.icr.io/cp/dcs/reports@sha256:a95d112e43f1ecf7f962f04354c64be6052319eb0c5cabf94c9af325648d9f46
  - name: cp.icr.io/cp/dcs/scaleafmdatamover@sha256:19aa20ce2b493f420e03758313e02f0351cf72836c15905f3c4b7285603e29c0
  - name: cp.icr.io/cp/dcs/scaleilmdatamover@sha256:725614999df502d09ba0621c4a2c317ea18818cb288ad38d0c46f98c78648ba9
  - name: cp.icr.io/cp/dcs/sdmonitor@sha256:2d853b11e533a9eee9e922034478289ed82d4641a16e9c797fafbaaff8fb1fdf
  - name: cp.icr.io/cp/dcs/tikaserver@sha256:11a8175a621e6a5fb093ff41d71eb20f8dca8a4186a24a843cc2713f634b382d
  - name: cp.icr.io/cp/dcs/uibackend@sha256:1bfd41169c4450852872b0f8d190d1b24b12a46040eb6a167437a2f8086c798f
  - name: cp.icr.io/cp/dcs/uifrontend@sha256:58979c1c53c116e5f90e814166a129a1fa8587831f617147328b5b0a5843d3f1
  - name: cp.icr.io/cp/dcs/wkc-connector@sha256:3a68bf843669cce6f000ca14ecd5d42a6727fd8055de5ac157b9ecf360fd5c59
  - name: icr.io/cpopen/db2u-day2-operator@sha256:67e656ec5ad43ef62550852f42b346ee68b4f2e97867980e822819687aa740ba
  - name: icr.io/cpopen/db2u-day2-operator@sha256:6e87d78d96101ec3a8dfca7464e4196bd34b1bc700b7ef1b1de5add00a591e55
  - name: icr.io/cpopen/db2u-day2-operator@sha256:af106b46b923745688ef2952da39add7d38b46768e18bf2e2147b4b05e755f00
  - name: icr.io/cpopen/db2u-operator@sha256:9062cf50e0d1df382878bf3ad4d5c5cfdcdc801666e419cd3c6a62838445978c
  - name: icr.io/cpopen/db2u-operator@sha256:b838f643c872fee3552f889c7977d593438e6b9583b03798555f44737d601e82
  - name: icr.io/cpopen/db2u-operator@sha256:df5b2ee589a381253f8d0797885c8f798351914edf9e506cc1ff261ad3156f09
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:0f013b2188b1e44fd450b4753b66e38f0a3624cf7f0b1bdab01ab36105a5d08b
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:10e960e1e26d3b64ceda847618b757e1d7e162a3378668d1f85a14d1b38ec55a
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:9c3f363473b182a4291e2bf07c630bdc8907abb26b5ede6c36f55f3e07de6ac0
  - name: icr.io/cpopen/ibm-db2uoperator-catalog:253.0.0-26025-
hack@sha256:3d8a379ecc97328eee2e4d5c83f380c87f1b5a32c4097112116f24f287226830
  - name: icr.io/cpopen/ibm-db2uoperator-catalog:253.0.0-26025-
hack@sha256:d6b6fcd8be9359bc8854fb314a6a73b6b5018d3dda2a005a1d16f8b15b18d1a8
  - name: icr.io/cpopen/ibm-spectrum-discover-operator-
bundle@sha256:cced71d877089db30e7badc95a2113ae09136347777cae0ec49000a71beaac81
  - name: icr.io/cpopen/ibm-spectrum-discover-operator-catalog:253.0.0-26025-
hack@sha256:c8b42eafa039542123c4ba29ef650c4a41996a28fd4504fb7c4e2e62fedab5b1
  - name: icr.io/cpopen/ibm-spectrum-discover-
operator@sha256:0b97164be106bf8733de8e432c17afc9e597251c64d0a2c37f2da5edab70f9f1
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:1b0dab1db92c2a6c36ad24ac9d5aa52acde882dcbcff4e6f4ad8af4fa9b857b7
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:39baed72d1a73aa5a2fbf245d4fe20faa6762908acb6efe9659e6cc6bd3a5c6f
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:46e9621ee884f45907330610be5f5a4e4f5f978da2d35d4e0d77e70abd767386
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:4a34187ba2196012acfd2eab8fb06b07c679bc6c0c13fcc988676043afc1c968
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:67fb65c796d162a58648f2ede6a053e1b9fd76aab01c91aad9acf39c5afbb814
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:6a123d208d9936f9f0148588e944edc348da066570f11ba6627659acca79794a
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:9a9fcad406d1500f3b86f6820ca80b0fcf4ddc6b8e2b6796b60eb53e0c616a76
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:c93051ba7ba210c4ad1a4c904be3752355c5937daf10298a14fb02c098a24c71
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:fffb214cf777b845fa4a4617ff59a5b14d4ef94ca7e430b684bcbc3db6c75939
  - name: icr.io/db2u/db2u.db2wh.restricted.v2@sha256:6855ca6e06372a9072e248ccb1c5770b65400b1226859402411a9c18e0426789
  - name: icr.io/db2u/db2u.db2wh.restricted.v2@sha256:c6c98d491a01e10bf244e39b86d0073b3c3da4ccefaa15c15e9db562f7da0da7
  - name: icr.io/db2u/db2u.db2wh@sha256:0aa85bc1e6c8fe891fc4b3f2137d2afef611a52e11134055a76ce3d5602b6200
  - name: icr.io/db2u/db2u.db2wh@sha256:3cc0814e4850663815b34a21ba3d480bebe4d3120315c192ad33d1ef0d6b09
  - name: icr.io/db2u/db2u.db2wh@sha256:a21f1de1cf2accdd0c0e94f045cda34f1d2b6a0b292bce33e66fb0eb93fbf45b
  - name: icr.io/db2u/db2u.db2wh@sha256:da3dcced05f576ff140647b8fcf73bc2e8aad9d329cfb34d042350c487798caf
  - name: icr.io/db2u/db2u.dv.api@sha256:c775f04c1cf735e4eb9b654f8307e69a294d33a182310b2068c631365ace03e2
  - name: icr.io/db2u/db2u.dv.api@sha256:fe5d3b18469795b2bf9635a0d85b1b84542dc1506aaa5f3f5568620c1b11bc39
  - name: icr.io/db2u/db2u.dv.caching@sha256:16b671ca11da90cb2032f52e0adfdb11763137255b2f8abd75d0907c3c36ee6c
  - name: icr.io/db2u/db2u.dv.caching@sha256:52ab0a6385636334cd3f405b62aa5719c4b600e59ac1f26369773300c3d798297
  - name: icr.io/db2u/db2u.dv.utils@sha256:8327012b2fbe2789aee4855a54125dfd114572790d08054b205ae9f24d679587
  - name: icr.io/db2u/db2u.dv.utils@sha256:ed9da17c3676c6928572927d820b953be7b49074047ff94e61962e4fbd596b62
  - name: icr.io/db2u/db2u.hurricane@sha256:2e8c303ad6878d2de0b80c72f47140b93b25abdfe320aaf3e0de0380edf3ddb5
  - name: icr.io/db2u/db2u.hurricane@sha256:eb88a59e26bf023af2cf97c9fe162be817dcb2afc6a8fab053fc06989a243875
  - name: icr.io/db2u/db2u.instdb.restricted.v2@sha256:8f49d7dab7152d689d057297894174bd3e5ce604b5357d776223bc243fccfa84
  - name: icr.io/db2u/db2u.instdb.restricted.v2@sha256:b7112c671588766b31a889582ceef49557117c72a73ce0c94e53bd7739879e3c
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:1036e55662f721284a535bc63a46f1d8c61fa17ce2eb4c4b90f147f69ec35290
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:28982d3ea268ab42a865f96f6f93ad9e36602c4f8dc0a0cf87d29e0570bd7d40
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:56a9a4bde204ac715f6762704f263c6f384412d370a1bcfd629f91a06995eed6
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:751cdc1796099c74ced03bda277dd1950166255d4aa0410a7f21e94276c68139
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:76ab047ae55cc5f1d34c46cbab64e33c725d38f93a9d74c62eb5426521718a99
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:a2a2edd7329585d5788ddd3e02865ccc9f2f29a81430e11bb9278af4c5b9eff9
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:ae828eeb870bc285298c3647c8767de790e36763984702db9621e94c9985e17e
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:df1e5e85210883d5a4586eca57d57182cb5a0974cc23cb5b4cd66aea2a2e749b
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:f22f0ab6ba02dfbdb353241250e7c4bb2e834b6f7dd89c88abe502e96916ff15
  - name: icr.io/db2u/db2u.instdb@sha256:1679240cfce999a2c4b4cac5371ad2c08784223acafd009347837498bd60315f
  - name: icr.io/db2u/db2u.instdb@sha256:275ac3b1b86238c80b9ac66619d868a7ece1a2c556198f5b5fc59a9b8eb84dc3
  - name: icr.io/db2u/db2u.instdb@sha256:6a12e1dda62f990cef4fd71cd5586dc724a2bd9c494e16b6cae9c89043c9a43b
  - name: icr.io/db2u/db2u.instdb@sha256:6b4cede07d71895f4f1983c1e7c09fef75324ec05b0ed36febe1f64a1b4773cd
  - name: icr.io/db2u/db2u.instdb@sha256:969417d181b532fbdb6853bfd15dfd391b44685d540661c53441f54d7d1d8f0b
  - name: icr.io/db2u/db2u.instdb@sha256:bff81001e977b2e4817fd998ef35c862a2fcba3b3ef80d29c8fc5d905e02fe81
  - name: icr.io/db2u/db2u.instdb@sha256:d5641eebf385829def0e5c667a0694b982f5d7c6460600e70c865f6fafb18de2
  - name: icr.io/db2u/db2u.instdb@sha256:dcd21a12aa9406313619fa05833e80cd64a52baa902cdde825300bb1a2b69b75
  - name: icr.io/db2u/db2u.instdb@sha256:e65987d494390105e9a9982b915ef362cbe5ee289582ce101063785ed16f2d05
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:13cdb4161aba7c9d2ef65a1eefd1131fb661bf0edbb05d475186280bd2661768
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:16b0a2fe8046660194792392470ebb5a6a18e82d5852ee31fe2f71234f1712c
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:408e13e0a250bd7daa1fe68f293dd5d22153ebdce8be409257bb3fec8f4406c1
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:5e80225570f5664873555adeb9ed20ee924d23f8719270a704f38956a1bbe8b9a
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:63d959334719870cf211e44897af2d05e5b5fb9d65daa4b86c48db51582a423ee
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:a9e3ff99fa60555805085505b3ecaae9be3371ae17ece47db6edfcedb0b2dcf5
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:bff1c6828d113f0997ebad884fadcb3953653ec29714cf81643b6f669e190fc8
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:c46a7d08092cb9593f8f6473769d1de04533ea04fdadf3093fc51c59bca3500d
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:e1057188f633d8ccc6b8ca3c6830f3d71266f5998e5800f9fe572529d4778b77
  - name: icr.io/db2u/db2u.mustgather@sha256:01951471547ee1ce3190fc2c03ada0f47623b7928b184d22173d06aeb458d73b
```

```
  - name: icr.io/db2u/db2u.mustgather@sha256:09268f018b77b94e8536c0f1aabec5fc8f5610c14bf31bedc59e88cdc78177f2
  - name: icr.io/db2u/db2u.mustgather@sha256:11035b1812f9ae1792e5b45f0e4857d5914f638a6b060a66a0aa4fa5e1933fca
  - name: icr.io/db2u/db2u.mustgather@sha256:19f59de73fa2df4409ec57f0929f30820393f5ed1cfa6dc866fc2e7f3321ffe7
  - name: icr.io/db2u/db2u.mustgather@sha256:2a9e534ea4819b46d5fac2590251a07556155bade8aeb04ff5183e9f942ee31b
  - name: icr.io/db2u/db2u.mustgather@sha256:5a47466d9a0e43ea5c477fb66a84bd1d0f4e7f705ed2a5ff152730de6a7dbcaf
  - name: icr.io/db2u/db2u.mustgather@sha256:acbcca4773f79fe966a3685a24c9734481d2482e7c6137300b494b8bd12ab1cb
  - name: icr.io/db2u/db2u.mustgather@sha256:c100495d77c36755f882675600b48c39c3261609008d9512aa76a27397302ebd
  - name: icr.io/db2u/db2u.mustgather@sha256:db3d978cfa5553d0843e1320e9715b6d089c3e369007ed502459d2a1a2ba860d
  - name: icr.io/db2u/db2u.qrep@sha256:78b41c0f0ca28fd6b90b1aa26f7ed6089749d29c8c62d7698155dad7bc3582b5
  - name: icr.io/db2u/db2u.qrep@sha256:881cf716b9f2389fe59809d409f9f9f2ca687c6a665c0330edf208d34777e912
  - name: icr.io/db2u/db2u.qrep@sha256:a8069c11a1adcd3934a9ba3cdc596308ecdc84c301a735c6506287f398d01fbc
  - name: icr.io/db2u/db2u.qrep@sha256:c08a2f9d4ea71b44d0444b1f61ea898ba1d67fed2cf0023cf774dfed095107ae
  - name: icr.io/db2u/db2u.qrep@sha256:e14893f0d00a8d06be5a9476dd1f3451957c454b7567fa30bfba4d9432b6a34d
  - name: icr.io/db2u/db2u.qrep@sha256:ed535873bb8606cef4bc6d87d036268647ab303e45a5849b3c3d03226b7dc6fb
  - name: icr.io/db2u/db2u.rest@sha256:0c41319691497cb7793cd91b53bb3e7084ea02ece9589d830ba5d130852125cc
  - name: icr.io/db2u/db2u.rest@sha256:1352b0fb4550dd13e4bb08e58832f6c203eba993950166dd9f003a9e8c162432
  - name: icr.io/db2u/db2u.rest@sha256:209edf7fa0c155efd7e4c209d68abed1747f8a02ba340f7015f0ee436da7ebd7
  - name: icr.io/db2u/db2u.rest@sha256:5cb5f6e25c4f6a789d1680478c266affe7719964faa16328822621c5f43b0cce
  - name: icr.io/db2u/db2u.rest@sha256:681ace912b6df4732c6b364f585a6782fcacb102e0465cb0df44e7417ee6a9f0
  - name: icr.io/db2u/db2u.rest@sha256:cd57827e98bca2488dc991c65276fe125e3a32f5c2fa95df1151e3deb9bf412f
  - name: icr.io/db2u/db2u.rest@sha256:d04710c5df92796e864da8cdc403b9ee2fa8bf8f006d6cc8f0cb59340f8163c3
  - name: icr.io/db2u/db2u.rest@sha256:dc657410c33beef847ffadc8e47f897fdf3c614800cddc7c52e48dcb7cacea09
  - name: icr.io/db2u/db2u.rest@sha256:f2e19e2a144c7e852baf4150e0668bdef37b9ea30c1ca7da773467ed780fb5f6
  - name: icr.io/db2u/db2u.restricted.v2@sha256:1788f4876b7153f51b28ca9307748d0f83ce4e42cd55ff01e29f76adc6bb29eb
  - name: icr.io/db2u/db2u.restricted.v2@sha256:df99b606a2b1683b2b064f1191e651f1df2c64458169de8f4cd14e1d10805876
  - name: icr.io/db2u/db2u.restricted@sha256:0c5b47350e0189147104bc5dd273803868608ab048f78783c6f640f4419e89eb
  - name: icr.io/db2u/db2u.restricted@sha256:20a17afd17d48c80b6129b8bcbfce7117a5fee909c44d27603cd6c2807b2dfc8
  - name: icr.io/db2u/db2u.restricted@sha256:565ddd470867984d282b26873c1440a0dd96b664a1f4f169b17a1b76065dadf8
  - name: icr.io/db2u/db2u.restricted@sha256:6999dba7f8c1c348dc79d81bbfcc468096db6d3ef7acb01facd3b60108807d9d
  - name: icr.io/db2u/db2u.restricted@sha256:7698c04b23be8058facc1d2dee69b6a94d9f95db9e25de3eafd9fac2622f8d25
  - name: icr.io/db2u/db2u.restricted@sha256:8a207ff2da1a29cae1c9d7ef521ca27a3f30330d43a7c351a379f93e178287ea
  - name: icr.io/db2u/db2u.restricted@sha256:8f89f5476a4d42ed558a8d5fe39abb24c75152a79079fe6a462010017319d5b6
  - name: icr.io/db2u/db2u.restricted@sha256:9d73c208aca0a993487289905ce7932cfc46ab0b802aac213aeb0cb16ad44e80
  - name: icr.io/db2u/db2u.restricted@sha256:cae02720f1ed588e416e24c3d3af74253fd2b29a5160bf699e5db1943233aa25
  - name: icr.io/db2u/db2u.tools.restricted.v2@sha256:4dc23d46e8a535661e54ee57f9147b1dc21eef6fa70e07e0b009fcd932a3d505
  - name: icr.io/db2u/db2u.tools.restricted.v2@sha256:78ab6edca7f3f725f1dc58d9010b0ccd3483df03ad8920ec7464c9c426e522b4
  - name: icr.io/db2u/db2u.tools@sha256:126301cb3616037133324c8c1352af0a7921226203a2ee9b9eeb736872e03311
  - name: icr.io/db2u/db2u.tools@sha256:22e995c30dae2dd792fcd0e07a7cf2b90fee23b67ac90b0722f11093c114e52a
  - name: icr.io/db2u/db2u.tools@sha256:38c70063a9700752b59d216a7ff7cfdb38fe4f40e345a091bd591feffe0ea2a5
  - name: icr.io/db2u/db2u.tools@sha256:67130a4444d5e5d5c9c2051f4983f0033a3a329aa1cb2e2488c06360f7134554
  - name: icr.io/db2u/db2u.tools@sha256:7e9f29fff47c0d8a7f4fe8feb4763cc3df5aa71b761434ccbd43d5a30fdaae3a
  - name: icr.io/db2u/db2u.tools@sha256:82ffd808fd4f316b6c4cbb327ab358cb0ceb1c110f12dd8a44ecc0264a33328d
  - name: icr.io/db2u/db2u.tools@sha256:b48f52abddec7da941c1613010936f9f651b70739194198914953cbb80992dae
  - name: icr.io/db2u/db2u.tools@sha256:b5815e23b5b60d4ee273ea52f4b752d1dc913464a29f59daee1eefd67cb6e87b
  - name: icr.io/db2u/db2u.tools@sha256:ba91e1d8780638b9871214449b8d76b2851c9d6e8c389990861df3669e2b198d
  - name: icr.io/db2u/db2u.update.image@sha256:0a0cc2a32b6ddff104722696b3799e43e94861c1f6af2d5c545b92aedf21cad5
  - name: icr.io/db2u/db2u.update.image@sha256:1a3c1e315216aee550aaa34f58b6f62c4ce274424afe3e5f569f083badbb5619
  - name: icr.io/db2u/db2u.update.image@sha256:2f4cb0440157f2199d6a0f73375dfa70953d2d22cd731984ddec761af2ed75b7
  - name: icr.io/db2u/db2u.update.image@sha256:5ba422423bc4ec0ab4bf6e044038fb71e41f51b9bd958b40aa4c54b807a6d45b
  - name: icr.io/db2u/db2u.update.image@sha256:770b560ba8e16a0be7b07edc11c41fd46ed0ad849059199fea207569d49ef953
  - name: icr.io/db2u/db2u.update.image@sha256:83a9f12d93fba82b7015819fd0baffa68e83644cf903c3df0d0c535141a8669e
  - name: icr.io/db2u/db2u.update.image@sha256:8e547fc1968b187a28187a4b998f3f70ebdeeb0aefc4938ee058db8588eae421
  - name: icr.io/db2u/db2u.update.image@sha256:a8ec7ac89729d73195459bb6a55b7151f4edb66c93fc4ad3aaf51c942116629d
  - name: icr.io/db2u/db2u.update.image@sha256:bdb4840d7fb72a69471da736704fa6968341182b5447ebb31a5c6a9223ac440b
  - name: icr.io/db2u/db2u.veleroplugin@sha256:2891904a3dcf7fbdcddcf48fe78bc00a7f31e2604716e99011114bb3d1ccf40d
  - name: icr.io/db2u/db2u.veleroplugin@sha256:4096b0785d1fa5019dea77017c47fb0d4ad32fceef2ffc23c88095c630bb6d45
  - name: icr.io/db2u/db2u.veleroplugin@sha256:6905ab5396f6b86c0eb70513b9b5f871c4d3832efee7694e8187451b4d8115a4
  - name: icr.io/db2u/db2u.veleroplugin@sha256:6e3141301930950e3a0671dd8d74a7148a7ef790284e607fd3d6870edcbd823a
  - name: icr.io/db2u/db2u.veleroplugin@sha256:73d8c40d160da82b2218b40ac5e1b065339db8d7bf6914ac694ebba071a858ad
  - name: icr.io/db2u/db2u.veleroplugin@sha256:7a890053b98b6e510f9697175071bc5bae00cdd1ea671c2949dfec9a61830403
  - name: icr.io/db2u/db2u.veleroplugin@sha256:ce73ee327f940e12bc22e20324d72acd7266f4adf730469d4fbbec482481f8e3
  - name: icr.io/db2u/db2u.veleroplugin@sha256:d327a193a266c982c30bc009a8def0c84d75dd4f28bf3806bee82e8b173946a0
  - name: icr.io/db2u/db2u.veleroplugin@sha256:f4f2e9e3afdcaa7a9e4e522bcdb2aaa6b79db97361968bdf1428961b1ada878d
  - name: icr.io/db2u/db2u.watsonquery@sha256:33d6ee3007e62284cf7852cfd337393a0bfb86432df599fd971c31836abaa28b
  - name: icr.io/db2u/db2u.watsonquery@sha256:48cce86af50ba003dbed9ce54b0240b9eec59a2702ab93bae58a649d7fe777ce
  - name: icr.io/db2u/db2u@sha256:0ecd06074f6ce0f090f93b2b5ed51e41679154d638dc2031d64498aab2c0de68
  - name: icr.io/db2u/db2u@sha256:323e043a92970faa921733dbd4d474bab3e80dc105fc6fe4b9c5e0d2160e1856
  - name: icr.io/db2u/db2u@sha256:64d1fc881fa7c4f9203a1e43b498464333768623cc1494d42e2d8aac21b4af3c
  - name: icr.io/db2u/db2u@sha256:9678623a716ceb4ed9367812b842cd56227ed15858edae08f7bcd5961dfa5e2d
  - name: icr.io/db2u/db2u@sha256:b8701f8d719345ee156c2490d8a0fc9ee08a2cfca7c4537d728c729337da9b77
  - name: icr.io/db2u/db2u@sha256:d282a83ba1d016035d9e7c8f489b3599cc3e39826890e927856a5232b6f6013a
  - name: icr.io/db2u/db2u@sha256:d7c92b368064d15ddf6bcc3d96dc19f876603f5a9fb453185724607e2b966567
  - name: icr.io/db2u/db2u@sha256:ef563ca49bf4ad388e8af360a5a1d87fd756d0e9329c6a181f3a37c29429b9c6
  - name: icr.io/db2u/db2u@sha256:f95b4ad43756f1f256df6f9b8cc79757db430de06a0fea22b78261490a73649a
  - name: icr.io/db2u/etcd@sha256:3c5d8ff1c77f146b6db7224bcfc92d6d64a832332cc087a162b2cda81ea99eef
  - name: icr.io/db2u/etcd@sha256:84dda9076ae58f2f02abba6102ab8325e4c301984b2588cb751e46245c636940
  - name: icr.io/db2u/etcd@sha256:9fa13ffeab820c5a62bc5bdb41347ef5cfd5244584117f0f3bd28cfe8690d358
  - name: icr.io/db2u/etcd@sha256:a5123f1a68e08bbda547eb653b724b917c9ec4d4cb6f447b066c4b71c6828909
  - name: icr.io/db2u/etcd@sha256:af1d052b60f599452ecbb44a060173922286e296cbf52d5a551a485cc28900e6
  - name: icr.io/db2u/etcd@sha256:b211cdc9c094ecad0f8bb3d0b714331937bb26582e450b3babec158f9857976a
  - name: icr.io/db2u/etcd@sha256:b617959f55b413b22a5bb72a80c19c7ff966a695dd4793903c9044165296aa3c
  - name: icr.io/db2u/etcd@sha256:cc1f51f48d7d95bd25a1658490adbdc0b7d8d0429e23ff621b3b8b457d51e271
  - name: icr.io/db2u/etcd@sha256:e386ff2ca65b99c63c6166de9a649875891f493091754aa4dc0e83e92dae7cb2
  blockedImages:
  - name: icr.io/cpopen/db2u-day2-operator@sha256:91a7ebbbf0667cf18e6686145dcb9acf2df2d616ee61073309100247e3b7e6c3
  - name: icr.io/cpopen/db2u-operator@sha256:4a938eb64e44e5bf155293e1939c4c8f2bcee9603647167994a94dbd79ced316
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:8fc35a4bdb7db8231f0258986a3dcad014b80059681a55ce4cd2a87659c52849
  - name: icr.io/cpopen/ibm-db2uoperator-catalog:253.0.0-26025-
hack@sha256:3325c4e4ec118c186f80b8336e5a3a5d8e2c2d7a1f06d97c9fee1cc4c90d68dc
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:12713664b4753551ae3ac303c25c4df938a4394e241e1f9bbc526e74a8ad562e
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:4b890828cc50851a42e5fec6bb929d1f71578aeb143794cbfaa1207eb0d7d048
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:890d04affc2beb381a483162aa3e998599b308ca1615ecd63e66ba04d94c224a
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:b711c164a62ccd4db955ea67afb0e225601429b6406d98fbad31c5fea13a86ea
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:d35b06c1d03707642c6641aff7d481be128064b3aabb384e5aac43715da45d04
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:e57279502a5d605931494771e8de0b382d3ddb1a5d744820f70977bb777bdaf9
  - name: icr.io/db2u/db2u.db2wh@sha256:1594796ebbe52b158c12eeb8e5d4faa791f6c9a28f96e33844a05f0282b4fc1f
```

```
  - name: icr.io/db2u/db2u.db2wh@sha256:394a2a0c640a884c1f63332bce1a8d5ad73baba0d3b3a05b4fa876e35bf9d82e
  - name: icr.io/db2u/db2u.dv.api@sha256:f8ed8a89397c5aa2d6b61ac2ed9d55e3b6351c3c0396355c17d7e44a48038def
  - name: icr.io/db2u/db2u.dv.caching@sha256:d5fe8bbd87c5d4e4424f5bd71ae3bf35769ea744322b152128ab563412eb68b0
  - name: icr.io/db2u/db2u.dv.utils@sha256:83ec4f59d30685a3ab43c74ee20336a16635935f339b78c4744058504e110f5e
  - name: icr.io/db2u/db2u.hurricane@sha256:7e7d7b1df0122120aeb33864b00cd5b9d8a2beab0ee6e5bbe273b0b7ea4f0e14
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:22bb255ac34cff1d99255a52a94b6560880f5f51937fdfcedba1ccd668688d9c
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:61c8a176b449e0aa7638a524297a80ece4ec52700d99299ae855db5b54609438
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:7906196d7910eada7f94bb375f4d6d7e5174992a6a0b3aa3308ce12283df45a8
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:7ea95db7c6249ad9b115c5934b4849f68b633024db937bb7835ec3cb4ab6a897
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:b67acd5ee75833ec1e10705a0c4e7ba0f4be85ab03ed37527b56250ee3448cb9
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:d5e452c1f3f5dcacb747713e56dcd006be6ba291bbf305352cd411a9bae57ee6
  - name: icr.io/db2u/db2u.instdb@sha256:4e8688e4964cbbeb4a5329294a0d995a0a45a9bec237e3f0a18491f0bce97784
  - name: icr.io/db2u/db2u.instdb@sha256:5ad3fc61b531781b212275eed3c3632ba0b7cd7d5cb48cf321b3f2308b66c421
  - name: icr.io/db2u/db2u.instdb@sha256:829a95722e17914869a4e224f35ace529607e71a1d2235f6e6fe7f33378d1cb5
  - name: icr.io/db2u/db2u.instdb@sha256:a6c008e42a208fe1407cf8e9a9c5e4d97eb923aee7269821fad9bd037b407357
  - name: icr.io/db2u/db2u.instdb@sha256:cc300234190d53a01cbe7349b628df0518553b4cd6795bebfbbacd66ca5ae29b
  - name: icr.io/db2u/db2u.instdb@sha256:d1f0a4cd8591bc509208ab01d0268881d4a8529d53bae147e570de5a0ead3bb1
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:344840dc5a825a755f9701792bfc5f8e7766b776665c88e83a04bec8a1cd85ac
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:7313133d3b83dbf4a214c73e809b21b431bf81f4ff3c027a4ebe0608c254c9fc
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:7e7ac8e3f66e45c99e4fcf8673e09b18e2c0f7a0d6f72c3f86eb2eb157fabaf8
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:88574364d46b83ca98f20569e7ba0533fd38dd6a3cdff23bc374a629de1379e3
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:b8288381e8b2538464097db4328641648a44c5e9671bbc502af2e997b39abb41
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:f84b460effa534c060592012f01d85f16d340557931e3532f5f6e0b4b892b266
  - name: icr.io/db2u/db2u.mustgather@sha256:51707eef34bad0eeeecfb09c5746782851a0b4d6af562b626dc667ec5e50c723
  - name: icr.io/db2u/db2u.mustgather@sha256:517ae4be20211a68c11d5848f6330ec156bff90c3c64057d273b52d4bb98dae9
  - name: icr.io/db2u/db2u.mustgather@sha256:536bf34b07defd3bc624a7a3973e6d5e701d5fdb949978cfcb1dbd546968c475
  - name: icr.io/db2u/db2u.mustgather@sha256:628ac4c7d5e9b6f527cf451f397da5a8723956a283fa9a3dfe1db2dfc91bc858
  - name: icr.io/db2u/db2u.mustgather@sha256:6c91db906f8dd87bef288595ef2393d910f124c8d31440731af8b260fd2fa55d
  - name: icr.io/db2u/db2u.mustgather@sha256:f0a3a3dd13c7d133d39b25234977a150993334ae45049cb51a7fffc15943cd9a
  - name: icr.io/db2u/db2u.qrep@sha256:3c731b912f43a68b8ab8fef2bd61a4fe5287fac5c251046a160a049730f60f5f
  - name: icr.io/db2u/db2u.qrep@sha256:44724bda2d5336f17bec950e40195f4cb6199291a22a3c6d3d9b47407e0155f1
  - name: icr.io/db2u/db2u.qrep@sha256:9512f41becbcfc6ebf8afcaf35521478fd87e2b2881d856d443b5bb882cb8d79
  - name: icr.io/db2u/db2u.qrep@sha256:d6bcb17bd3ff2776ee3588ddebd946cade42a76f38666ccfc425635b09a6eb12
  - name: icr.io/db2u/db2u.rest@sha256:6e1f90125ad8a4740bf388d16f44e73737379f4dd9b881bc2559aea2d60d9976
  - name: icr.io/db2u/db2u.rest@sha256:778a3bd330aa7c9aa7cf405551856e92d3d8b40d75a509054d8c115e73e7b795
  - name: icr.io/db2u/db2u.rest@sha256:a12723d1671e2321286c22cfad30352c7860cd95f8099e45b4847036ccc0830e
  - name: icr.io/db2u/db2u.rest@sha256:a8158630c0ad0a21592a9e08891035d0d02d7b406c15f4740db77fec125b03ce
  - name: icr.io/db2u/db2u.rest@sha256:bf6576eecd916419d2d521289970094e0c73ded5de5154030678d6543c60a2de
  - name: icr.io/db2u/db2u.rest@sha256:d000f87783bf8963423064de28d069e6877aed8ec94d572c6b22a27313fec8ff
  - name: icr.io/db2u/db2u.restricted@sha256:27e1777bf955d26ce713cb8d6ffc6ee1d8d76c79dcbe94c07d1ec154b12519bd
  - name: icr.io/db2u/db2u.restricted@sha256:33a70ec4786221efe5e5fd397951111fc90ea3b75842ef9558c889034564e91b
  - name: icr.io/db2u/db2u.restricted@sha256:4e4397c45c10f09f723ff11b3b51bb7adb492898d652501abd85511cc660ccc6
  - name: icr.io/db2u/db2u.restricted@sha256:6b204a6026e176bc7c7404fd41ede9399439cf5347698c95b838a78454ace6d9
  - name: icr.io/db2u/db2u.restricted@sha256:9b388eab7b17d543bf9c731a2583bd2606ba6199873ab914a3cdd2065e231385
  - name: icr.io/db2u/db2u.restricted@sha256:c75c667f44a6ab63447d0e72ad2743beb6f34f29f0cdee9b83d83c72ebb2457e
  - name: icr.io/db2u/db2u.tools@sha256:02f1771f4a00db39a5ef741dc39eb84caedfd7393092607994500d9b481b9a54
  - name: icr.io/db2u/db2u.tools@sha256:35ae6c6618b443c49a17ff32f5ba0c098e46fc542482c02e94857eb378ace419
  - name: icr.io/db2u/db2u.tools@sha256:83c657db8954d116a97daae90957ed8b31a89055913de08c92bbe5d30d3c71ec
  - name: icr.io/db2u/db2u.tools@sha256:adeffc101962a8181b4ceba587d046808c6440ca4a516cabdb04f973b5e299e9
  - name: icr.io/db2u/db2u.tools@sha256:e89e56844720d8512e0e0ff65a0331719bc4c203cabf2eb1da329a6dd3d00c70
  - name: icr.io/db2u/db2u.tools@sha256:f22e5bbc28263a51aa45c4a42dfaabee793322594c32a58a13914b977aca300f
  - name: icr.io/db2u/db2u.update.image@sha256:318c6f072f745c191feecb34576bb68681dd9db3644cc32191fba34274332852
  - name: icr.io/db2u/db2u.update.image@sha256:56ec63418fba021fb1eafe54bc6d62106dfa9819a3ce88707fabcadb98b239aa
  - name: icr.io/db2u/db2u.update.image@sha256:5af408ae4cb77592ee6726abe3a281b9a7fac500231d05b45b0a2b05ed88293d
  - name: icr.io/db2u/db2u.update.image@sha256:836a33719793c32744c0f830890633cd1648f282585ae672d3067bd146b1d467
  - name: icr.io/db2u/db2u.update.image@sha256:9a5875da2221ed1e2725a085ec7abd2e0418379686e337987f9496367e7c7a61
  - name: icr.io/db2u/db2u.update.image@sha256:d327bb0e5cc2a453d2fad23b629ba4862b5af9f794fc7f10ecf9d46e5d107f19
  - name: icr.io/db2u/db2u.veleroplugin@sha256:1a2d4657bf14fa60ffa1b0dbca50b696f1ff4ffd28f4c470db67b0ecca5d6fb4
  - name: icr.io/db2u/db2u.veleroplugin@sha256:51114e7afaa01a9002289017a22a2f5c9c0eceeb8f2df857c8148dbb5651e46b
  - name: icr.io/db2u/db2u.veleroplugin@sha256:796518c555504dd1e5b77504ca07397c114ac6753adbf8c8de41dcd2e1b57065
  - name: icr.io/db2u/db2u.veleroplugin@sha256:a2b0b6ff986b618fe880acec97d4af1673f4bffda385e05b6e14301c1b51d472
  - name: icr.io/db2u/db2u.veleroplugin@sha256:b3ecbc09923c288fa8d21d9e36a23a61cbec71a0c43bdb1e8213688bab571626
  - name: icr.io/db2u/db2u.veleroplugin@sha256:f87644162951b7210e450f567dacf718d3480a2f3468b36131b7a43286a6c44f
  - name: icr.io/db2u/db2u.watsonquery@sha256:897237498f224723b5818c50cd37d218032c4a05ebe2bda15bebe99d13ef0521
  - name: icr.io/db2u/db2u@sha256:01bdc550ea7d727fa5715f96c61f7ffe763ee38fc1875132187c40e09d08a6aa
  - name: icr.io/db2u/db2u@sha256:427657ba472b22f66b68d2a4f2ea7564572853083508ae2a8dbf770d1a181b85
  - name: icr.io/db2u/db2u@sha256:4319988ff6a8dff6a97368f253fed5cb268eb0c20de4db9dedd77cc47214f806
  - name: icr.io/db2u/db2u@sha256:6dda7e511b92ffc83c4b92e7b5ed99b5b619a5986af4a86d75c5f4425917dd91
  - name: icr.io/db2u/db2u@sha256:94226fa2a0fa69f8438265f481ffc05e830a75b9bb4a04f2a2b1f8d239758b09
  - name: icr.io/db2u/db2u@sha256:adc37d1fdfde105a516b8b891c494627703d90e9104181b2959a98e6630a4e94
  - name: icr.io/db2u/etcd@sha256:0c79ec0ad02a7e810ce4bc094fe85ad869ac973494161e85dbf32e394fcccf6c
  - name: icr.io/db2u/etcd@sha256:1a3362799bd9402aba7f99eed179c2747a520b37c994e63ad9e2b5fdeb4d5bd9
  - name: icr.io/db2u/etcd@sha256:1befb2b817dc6f7bf3f2fe3f5f2862b638d791d1284facddf789f245775e575c
  - name: icr.io/db2u/etcd@sha256:234c073c178f60ff8e8e29fdfda25b1e908ea0655ff61f3a4e29a51eefdea8eb
  - name: icr.io/db2u/etcd@sha256:6e0c097d8e6d9cdacc71f80312dd96b8a0d50a07ad48a333fce0bfc190f58553
  - name: icr.io/db2u/etcd@sha256:bb5c436b88c57715ffc991d0ab5a05979a4c092c65ad98296af2b1349926c45c
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v2alpha1
mirror:
  additionalImages:
  - name: cp.icr.io/cp/dcs/auth@sha256:6cb6e582d02fb017dbbc1c736f3409a08823fef56381747f13952bd8a613cabb
  - name: cp.icr.io/cp/dcs/connmgr-scheduler@sha256:cca24250b9bbfdfe8c6c85c613c5fa1a56d2b0fbbb0e48a4c56fc3f50ea338a2
  - name: cp.icr.io/cp/dcs/connmgr@sha256:cbefb44b2d448b36f90e4fca7973ab66bc1fbfd799eaa1c13b5185bf116f6f09
  - name: cp.icr.io/cp/dcs/db-schema@sha256:534c98417f909a248698281bc244ef1441b6a01dccf5a3055242c13956576c9a
  - name: cp.icr.io/cp/dcs/db2whrest@sha256:3233681ab2ec32515653f5659d6c02f8e12fbfddaa9d65747dfe07a62de1bfd2
  - name: cp.icr.io/cp/dcs/dcs-ai-mcp-server@sha256:d8af262ff80ee897797d322bbd8f8382cf4b823033901aea37d5531b27565365
  - name: cp.icr.io/cp/dcs/import-service@sha256:1003a8694eff91c163a72b88c19baeb5cd47a97d152ab64ac910b8b8b693f096
  - name: cp.icr.io/cp/dcs/import-tags-app@sha256:c37128ef0ad7f7add93c72c42c8c1902f75a2907e683af5d7af99d04ff4ea381
  - name: cp.icr.io/cp/dcs/isd-proxy@sha256:5056d6a265b2a37123806976c56722b86c5c20dad6d2f5933838c8d076b1e120
  - name: cp.icr.io/cp/dcs/keystone@sha256:b3128687e35f78bc25cefbe7254f625b5f01f344ae190d161cdbf79057109aa6
  - name: cp.icr.io/cp/dcs/llm-proxy@sha256:fce3aebc677eba45434f16039d3c678ccf4c1c71d709ab3b13b5aebd79c928b6
  - name: cp.icr.io/cp/dcs/metadata-extender@sha256:9a988643df596a1bbfb1094874273c3c1ca879e8a645677090409b4de1f37ac9
  - name: cp.icr.io/cp/dcs/mo-agent-extract@sha256:2f4d7b0d5ada12d39932f687cea031d88beb9cdac791b48cdca29ed23a9860ca
  - name: cp.icr.io/cp/dcs/mo-ubi-init@sha256:927bb277f576f2a772faf8dc9978a6592d5d08bfad3cdce81da180a9095a34c2
  - name: cp.icr.io/cp/dcs/moconsumer@sha256:2f1d3fdec099951cfc5f6788d4a4d944d66ec0aa8e39c9688239f1ecad38729f
  - name: cp.icr.io/cp/dcs/moproducer@sha256:c984c05bc47939e22cec6f4c8c0a9f365b71b7582c3711a36a59b96cc546da78
```

```
  - name: cp.icr.io/cp/dcs/must-gather@sha256:3f655e31efb04862c52333f113322a9169ea54d88f6271777074 53fd48acec50
  - name: cp.icr.io/cp/dcs/policyengine@sha256:ffe2fac7ee8bd49be48a142818ca760c6482018d9a79941296ef8d3424c1949c
  - name: cp.icr.io/cp/dcs/remote-content-search@sha256:7c0a39bcf04c1063fffc866ec8513066beb398dfd28cdb436b21cf7c4754f7c8
  - name: cp.icr.io/cp/dcs/reports@sha256:a95d112e43f1ecf7f962f04354c64be6052319eb0c5cabf94c9af325648d9f46
  - name: cp.icr.io/cp/dcs/scaleafmdatamover@sha256:19aa20ce2b493f420e03758313e02f0351cf72836c15905f3c4b7285603e29c0
  - name: cp.icr.io/cp/dcs/scaleilmdatamover@sha256:725614999df502d09ba0621c4a2c317ea18818cb288ad38d0c46f98c78648ba9
  - name: cp.icr.io/cp/dcs/sdmonitor@sha256:2d853b11e533a9eee9e922034478289ed82d4641a16e9c797fafbaaff8fb1fdf
  - name: cp.icr.io/cp/dcs/tikaserver@sha256:11a8175a621e6a5fb093ff41d71eb20f8dca8a4186a24a843cc2713f634b382d
  - name: cp.icr.io/cp/dcs/uibackend@sha256:1bfd41169c4450852872b0f8d190d1b24b12a46040eb6a167437a2f8086c798f
  - name: cp.icr.io/cp/dcs/uifrontend@sha256:58979c1c53c116e5f90e814166a129a1fa8587831f6171473288b5b0a5843d3f1
  - name: cp.icr.io/cp/dcs/wkc-connector@sha256:3a68bf843669cce6f000ca14ecd5d42a6727fd8055de5ac157b9ecf360fd5c59
  - name: icr.io/cpopen/db2u-day2-operator@sha256:67e656ec5ad43ef62550852f42b346ee68b4f2e97867980e822819687aa740ba
  - name: icr.io/cpopen/db2u-day2-operator@sha256:6e87d78d96101ec3a8dfca7464e4196bd34b1bc700b7ef1b1de5add00a591e55
  - name: icr.io/cpopen/db2u-day2-operator@sha256:af106b46b923745688ef2952da39add7d38b46768e18bf2e2147b4b05e755f00
  - name: icr.io/cpopen/db2u-operator@sha256:9062cf50e0d1df382878bf3ad4d5c5cfdcdc801666e419cd3c6a62838445978c
  - name: icr.io/cpopen/db2u-operator@sha256:b838f643c872fee3552f889c7977d593438e6b9583b03798555f44737d601e82
  - name: icr.io/cpopen/db2u-operator@sha256:df5b2ee589a381253f8d0797885c8f798351914edf9e506cc1ff261ad3156f09
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:0f013b2188b1e44fd450b4753b66e38f0a3624cf7f0b1bdab01ab36105a5d08b
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:10e960e1e26d3b64ceda847618b757e1d7e162a3378668d1f85a14d1b38ec55a
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:9c3f363473b182a4291e2bf07c630bdc8907abb26b5ede6c36f55f3e07de6ac0
  - name: icr.io/cpopen/ibm-db2uoperator-catalog:253.0.0-26025-
hack@sha256:3d8a379ecc97328eee2e4d5c83f380c87f1b5a32c4097112116f24f287226830
  - name: icr.io/cpopen/ibm-db2uoperator-catalog:253.0.0-26025-
hack@sha256:d6b6fcd8be9359bc8854fb314a6a73b6b5018d3dda2a005a1d16f8b15b18d1a8
  - name: icr.io/cpopen/ibm-spectrum-discover-operator-
bundle@sha256:cced71d877089db30e7badc95a2113ae09136347777cae0ec49000a71beaac81
  - name: icr.io/cpopen/ibm-spectrum-discover-operator-catalog:253.0.0-26025-
hack@sha256:c8b42eafa039542123c4ba29ef650c4a41996a28fd4504fb7c4e2e62fedab5b1
  - name: icr.io/cpopen/ibm-spectrum-discover-
operator@sha256:0b97164be106bf8733de8e432c17afc9e597251c64d0a2c37f2da5edab70f9f1
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:1b0dab1db92c2a6c36ad24ac9d5aa52acde882dcbcff4e6f4ad8af4fa9b857b7
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:39baed72d1a73aa5a2fbf245d4fe20faa6762908acb6efe9659e6cc6bd3a5c6f
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:46e9621ee884f45907330610be5f5a4e4f5f978da2d35d4e0d77e70abd767386
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:4a34187ba2196012acfd2eab8fb06b07c679bc6c0c13fcc988676043afc1c968
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:67fb65c796d162a58648f2ede6a053e1b9fd76aab01c91aad9acf39c5afbb814
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:6a123d208d9936f9f0148588e944edc348da066570f11ba6627659acca79794a
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:9a9fcad406d1500f3b86f6820ca80b0fcf4ddc6b8e2b6796b60eb53e0c616a76
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:c93051ba7ba210c4ad1a4c904be3752355c5937daf10298a14fb02c098a24c71
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:fffb214cf777b845fa4a4617ff59a5b14d4ef94ca7e430b684bcbc3db6c75939
  - name: icr.io/db2u/db2u.db2wh.restricted.v2@sha256:6855ca6e06372a9072e248ccb1c5770b65400b1226859402411a9c18e0426789
  - name: icr.io/db2u/db2u.db2wh.restricted.v2@sha256:c6c98d491a01e10bf244e39b86d0073b3c3da4ccefaa15c15e9db562f7da0da7
  - name: icr.io/db2u/db2u.db2wh@sha256:0aa85bc1e6c8fe891fc4b3f2137d2afef611a52e11134055a76ce3d5602b6200
  - name: icr.io/db2u/db2u.db2wh@sha256:3cc0814e4850663815b34a21ba3d480bebe4d312031315c192ad33d1ef0d6b09
  - name: icr.io/db2u/db2u.db2wh@sha256:a21f1de1cf2accdd0c0e94f045cda34f1d2b6a0b292bce33e66fb0eb93fbf45b
  - name: icr.io/db2u/db2u.db2wh@sha256:da3dcced05f576ff140647b8fcf73bc2e8aad9d329cfb34d042350c487798caf
  - name: icr.io/db2u/db2u.dv.api@sha256:c775f04c1cf735e4eb9b654f8307e69a294d33a182310b2068c631365ace03e2
  - name: icr.io/db2u/db2u.dv.api@sha256:fe5d3b18469795b2bf9635a0d85b1b84542dc1506aaa5f3f5568620c1b11bc39
  - name: icr.io/db2u/db2u.dv.caching@sha256:16b671ca11da90cb2032f52e0adfdb11763137255b2f8abd75d0907c3c36ee6c
  - name: icr.io/db2u/db2u.dv.caching@sha256:52ab0a6385636334cd3f405b62aa5719c4b600e59ac1f2636977300c3d798297
  - name: icr.io/db2u/db2u.dv.utils@sha256:8327012b2fbe2789aee4855a54125dfd114572790d08054b205ae9f24d679587
  - name: icr.io/db2u/db2u.dv.utils@sha256:ed9da17c3676c6928572927d820b953be7b49074047ff94e61962e4fbd596b62
  - name: icr.io/db2u/db2u.hurricane@sha256:2e8c303ad6878d2de0b80c72f47140b93b25abdfe320aaf3e0de0380edf3ddb5
  - name: icr.io/db2u/db2u.hurricane@sha256:eb88a59e26bf023af2cf97c9fe162be817dcb2afc6a8fab053fc06989a243875
  - name: icr.io/db2u/db2u.instdb.restricted.v2@sha256:8f49d7dab7152d689d05729789417 4bd3e5ce604b5357d776223bc243fccfa84
  - name: icr.io/db2u/db2u.instdb.restricted.v2@sha256:b7112c671588766b31a889582ceef49557117c72a73ce0c94e53bd7739879e3c
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:1036e55662f721284a535bc63a46f1d8c61fa17ce2eb4c4b90f147f69ec35290
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:28982d3ea268ab42a865f96f6f93ad9e36602c4f8dc0a0cf87d29e0570bd7d40
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:56a9a4bde204ac715f6762704f263c6f384412d370a1bcfd629f91a06995eed6
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:751cdc1796099c74ced03bda277dd1950166255d4aa0410a7f21e94276c68139
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:76ab047ae55cc5f1d34c46cbab64e33c725d38f93a9d74c62eb5426521718a99
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:a2a2edd7329585d5788ddd3e02865ccc9f2f29a81430e11bb9278af4c5b9eff9
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:ae828eeb870bc285298c3647c8767de790e36763984702db9621e94c9985e17e
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:df1e5e85210883d5a4586eca57d57182cb5a0974cc23cb5b4cd66aea2a2e749b
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:f22f0ab6ba02dfbdb353241250e7c4bb2e834b6f7dd89c88abe502e96916ff15
  - name: icr.io/db2u/db2u.instdb@sha256:1679240cfce999a2c4b4cac5371ad2c08784223acafd009347837498bd60315f
  - name: icr.io/db2u/db2u.instdb@sha256:275ac3b1b86238c80b9ac66619d868a7ece1a2c556198f5b5fc59a9b8eb84dc3
  - name: icr.io/db2u/db2u.instdb@sha256:6a12e1dda62f990cef4fd71cd5586dc724a2bd9c494e16b6cae9c89043c9a43b
  - name: icr.io/db2u/db2u.instdb@sha256:6b4cede07d71895f4f1983c1e7c09fef75324ec05b0ed36febe1f64a1b4773cd
  - name: icr.io/db2u/db2u.instdb@sha256:969417d181b532fbdb6853bfd15dfd391b44685d540661c53441f54d7d1d8f0b
  - name: icr.io/db2u/db2u.instdb@sha256:bff81001e977b2e4817fd998ef35c862a2fcba3b3ef80d29c8fc5d905e02fe81
  - name: icr.io/db2u/db2u.instdb@sha256:d5641eebf385829def0e5c667a0694b982f5d7c6460600e70c865f6fafb18de2
  - name: icr.io/db2u/db2u.instdb@sha256:dcd21a12aa9406313619fa05833e80cd64a52baa902cdde825300bb1a2b69b75
  - name: icr.io/db2u/db2u.instdb@sha256:e65987d494390105e9a9982b915ef362cbe5ee289582ce101063785ed16f2d05
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:13cdb4161aba7c9d2ef65a1eefd1131fb661bf0edbb05d475186280bd2661768
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:16b0a2fe8046660194799239247 0ebb5a6a18e82d5852ee31fe2f71234f1712c
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:408e13e0a250bd7daa1fe68f293dd5d22153ebdce8be409257bb3fec8f4406c1
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:5e80225570f566487355adeb9ed20ee924d23f8719270a704f38956a1bbe8b9a
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:63d95933471987cf211e44897af2d05e5b5fb9d65daa4b86c48db51582a423ee
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:a9e3ff99fa60555805085505b3ecaae9be3371ae17ece47db6edfcedb0b2dcf5
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:bff1c6828d113f0997ebad884fadcb3953653ec29714cf81643b6f669e190fc8
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:c46a7d08092cb9593f8f6473769d1de04533ea04fdadf3093fc51c59bca3500d
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:e1057188f633d8ccc6b8ca3c6830f3d71266f5998e5800f9fe572529d4778b77
  - name: icr.io/db2u/db2u.mustgather@sha256:01951471547ee1ce3190fc2c03ada0f47623b7928b184d22173d06aeb458d73b
  - name: icr.io/db2u/db2u.mustgather@sha256:09268f018b77b94e8536c0f1aabec5fc8f5610c14bf31bedc59e88cdc78177f2
  - name: icr.io/db2u/db2u.mustgather@sha256:11035b1812f9ae1792e5b45f0e4857d5914f638a6b060a66a0aa4fa5e1933fca
  - name: icr.io/db2u/db2u.mustgather@sha256:19f59de73fa2df4409ec57f0929f30820393f5ed1cfa6dc866fc2e7f3321ffe7
  - name: icr.io/db2u/db2u.mustgather@sha256:2a9e534ea4819b46d5fac2590251a07556155bade8aeb04ff5183e9f942ee31b
  - name: icr.io/db2u/db2u.mustgather@sha256:5a47466d9a0e43ea5c477fb66a84bd1d0f4e7f705ed2a5ff152730de6a7dbcaf
  - name: icr.io/db2u/db2u.mustgather@sha256:acbcca4773f79fe966a3685a24c9734481d2482e7c6137300b494b8bd12ab1cb
  - name: icr.io/db2u/db2u.mustgather@sha256:c100495d77c36755f882675600b48c39c3261609008d9512aa76a27397302e8bd
  - name: icr.io/db2u/db2u.mustgather@sha256:db3d978cfa5553d0843e1320e9715b6d089c3e369007ed502459d2a1a2ba860d
  - name: icr.io/db2u/db2u.qrep@sha256:78b41c0f0ca28fd6b90b1aa26f7ed6089749d29c8c62d7698155dad7bc3582b5
  - name: icr.io/db2u/db2u.qrep@sha256:881cf716b9f2389fe59809d409f9f9f2ca687c6a665c0330edf208d34777e912
  - name: icr.io/db2u/db2u.qrep@sha256:a8069c11a1adcd3934a9ba3cdc596308ecdc84c301a735c6506287f398d01fbc
  - name: icr.io/db2u/db2u.qrep@sha256:c08a2f9d4ea71b44d0444b1f61ea898ba1d67fed2cf0023cf774dfed095107ae
  - name: icr.io/db2u/db2u.qrep@sha256:e14893f0d00a8d06be5a9476dd1f3451957c454b7567fa30bfba4d9432b6a34d
```

```
  - name: icr.io/db2u/db2u.qrep@sha256:ed535873bb8606cef4bc6d87d036268647ab303e45a5849b3c3d03226b7dc6fb
  - name: icr.io/db2u/db2u.rest@sha256:0c41319691497cb7793cd91b53bb3e7084ea02ece9589d830ba5d130852125cc
  - name: icr.io/db2u/db2u.rest@sha256:1352b0fb4550dd13e4bb08e58832f6c203eba993950166dd9f003a9e8c162432
  - name: icr.io/db2u/db2u.rest@sha256:209edf7fa0c155efd7e4c209d68abed1747f8a02ba340f7015f0ee436da7ebd7
  - name: icr.io/db2u/db2u.rest@sha256:5cb5f6e25c4f6a789d1680478c266affe7719964faa16328822621c5f43b0cce
  - name: icr.io/db2u/db2u.rest@sha256:681ace912b6df4732c6b364f585a6782fcacb102e0465cb0df44e7417ee6a9f0
  - name: icr.io/db2u/db2u.rest@sha256:cd57827e98bca2488dc991c65276fe125e3a32f5c2fa95df1151e3deb9bf412f
  - name: icr.io/db2u/db2u.rest@sha256:d04710c5df92796e864da8cdc403b9ee2fa8bf8f006d6cc8f0cb59340f8163c3
  - name: icr.io/db2u/db2u.rest@sha256:dc657410c33beef847ffadc8e47f897fdf3c614800cddc7c52e48dcb7cacea09
  - name: icr.io/db2u/db2u.rest@sha256:f2e19e2a144c7e852baf4150e0668bdef37b9ea30c1ca7da773467ed780fb5f6
  - name: icr.io/db2u/db2u.restricted.v2@sha256:1788f4876b7153f51b28ca9307748d0f83ce4e42cd55ff01e29f76adc6bb29eb
  - name: icr.io/db2u/db2u.restricted.v2@sha256:df99b606a2b1683b2b064f1191e651f1df2c64458169de8f4cd14e1d10805876
  - name: icr.io/db2u/db2u.restricted@sha256:0c5b47350e0189147104bc5dd273803868608ab048f78783c6f640f4419e89eb
  - name: icr.io/db2u/db2u.restricted@sha256:20a17afd17d48c80b6129b8bcbfce7117a5fee909c44d27603cd6c2807b2dfc8
  - name: icr.io/db2u/db2u.restricted@sha256:565ddd470867984d282b26873c1440a0dd96b664a1f4f169b17a1b76065dadf8
  - name: icr.io/db2u/db2u.restricted@sha256:6999dba7f8c1c348dc79d81bbfcc468096db6d3ef7acb01facd3b60108807d9d
  - name: icr.io/db2u/db2u.restricted@sha256:7698c04b23be8058facc1d2dee69b6a94d9f95db9e25de3eafd9fac2622f8d25
  - name: icr.io/db2u/db2u.restricted@sha256:8a207ff2da1a29cae1c9d7ef521ca27a3f30330d43a7c351a379f93e178287ea
  - name: icr.io/db2u/db2u.restricted@sha256:8f89f5476a4d42ed558a8d5fe39abb24c75152a79079fe6a462010017319d5b6
  - name: icr.io/db2u/db2u.restricted@sha256:9d73c208aca0a993487289905ce7932cfc46ab0b802aac213aeb0cb16ad44e80
  - name: icr.io/db2u/db2u.restricted@sha256:cae02720f1ed588e416e24c3d3af74253fd2b29a5160bf699e5db1943233aa25
  - name: icr.io/db2u/db2u.tools.restricted.v2@sha256:4dc23d46e8a535661e54ee57f9147b1dc21eef6fa70e07e0b009fcd932a3d505
  - name: icr.io/db2u/db2u.tools.restricted.v2@sha256:78ab6edca7f3f725f1dc58d9010b0ccd3483df03ad8920ec7464c9c426e522b4
  - name: icr.io/db2u/db2u.tools@sha256:126301cb3616037133324c8c1352af0a7921226203a2ee9b9eeb736872e03311
  - name: icr.io/db2u/db2u.tools@sha256:22e995c30dae2dd792fcd0e07a7cf2b90fee23b67ac90b0722f11093c114e52a
  - name: icr.io/db2u/db2u.tools@sha256:38c70063a9700752b59d216a7ff7cfdb38fe4f40e345a091bd591feffe0ea2a5
  - name: icr.io/db2u/db2u.tools@sha256:67130a4444d5e5d5c9c2051f4983f0033a3a329aa1cb2e2488c06360f7134554
  - name: icr.io/db2u/db2u.tools@sha256:7e9f29fff47c0d8a7f4fe8feb4763cc3df5aa71b761434ccbd43d5a30fdaae3a
  - name: icr.io/db2u/db2u.tools@sha256:82ffd808fd4f316b6c4cbb327ab358cb0ceb1c110f12dd8a44ecc0264a33328d
  - name: icr.io/db2u/db2u.tools@sha256:b48f52abddec7da941c1613010936f9f651b707391941989149 53cbb80992dae
  - name: icr.io/db2u/db2u.tools@sha256:b5815e23b5b60d4ee273ea52f4b752d1dc913464a29f59daee1eefd67cb6e87b
  - name: icr.io/db2u/db2u.tools@sha256:ba91e1d8780638b9871214449b8d76b2851c9d6e8c389990861df3669e2b198d
  - name: icr.io/db2u/db2u.update.image@sha256:0a0cc2a32b6ddff104722696b3799e43e94861c1f6af2d5c545b92aedf21cad5
  - name: icr.io/db2u/db2u.update.image@sha256:1a3c1e315216aee550aaa34f58b6f62c4ce274424afe3e5f569f083badbb5619
  - name: icr.io/db2u/db2u.update.image@sha256:2f4cb0440157f2199d6a0f73375dfa70953d2d22cd731984ddec761af2ed75b7
  - name: icr.io/db2u/db2u.update.image@sha256:5ba422423bc4ec0ab4bf6e044038fb71e41f51b9bd958b40aa4c54b807a6d45b
  - name: icr.io/db2u/db2u.update.image@sha256:770b560ba8e16a0be7b07edc11c41fd46ed0ad849059199fea207569d49ef953
  - name: icr.io/db2u/db2u.update.image@sha256:83a9f12d93fba82b7015819fd0baffa68e83644cf903c3df0d0c535141a8669e
  - name: icr.io/db2u/db2u.update.image@sha256:8e547fc1968b187a28187a4b998f3f70ebdeeb0aefc4938ee058db8588eae421
  - name: icr.io/db2u/db2u.update.image@sha256:a8ec7ac89729d73195459bb6a55b7151f4edb66c93fc4ad3aaf51c942116629d
  - name: icr.io/db2u/db2u.update.image@sha256:bdb4840d7fb72a69471da736704fa6968341182b5447ebb31a5c6a9223ac440b
  - name: icr.io/db2u/db2u.veleroplugin@sha256:2891904a3dcf7fbdcddcf48fe78bc00a7f31e2604716e99011114bb3d1ccf40d
  - name: icr.io/db2u/db2u.veleroplugin@sha256:4096b0785d1fa5019dea77017c47fb0d4ad32fceef2ffc23c88095c630bb6d45
  - name: icr.io/db2u/db2u.veleroplugin@sha256:6905ab5396f6b86c0eb70513b9b5f871c4d3832efee7694e8187451b4d8115a4
  - name: icr.io/db2u/db2u.veleroplugin@sha256:6e31413019309 50e3a0671dd8d74a7148a7ef790284e607fd3d6870edcbd823a
  - name: icr.io/db2u/db2u.veleroplugin@sha256:73d8c40d160da82b2218b40ac5e1b065339db8d7bf6914ac694ebba071a858ad
  - name: icr.io/db2u/db2u.veleroplugin@sha256:7a890053b98b6e510f9697175071bc5bae00cdd1ea671c2949dfec9a61830403
  - name: icr.io/db2u/db2u.veleroplugin@sha256:ce73ee327f940e12bc22e20324d72acd7266f4adf730469d4fbbec482481f8e3
  - name: icr.io/db2u/db2u.veleroplugin@sha256:d327a193a266c982c30bc009a8def0c84d75dd4f28bf3806bee82e8b173946a0
  - name: icr.io/db2u/db2u.veleroplugin@sha256:f4f2e9e3afdcaa7a9e4e522bcdb2aaa6b79db97361968bdf1428961b1ada878d
  - name: icr.io/db2u/db2u.watsonquery@sha256:33d6ee3007e62284cf7852cfd337393a0bfb86432df599fd971c31836abaa28b
  - name: icr.io/db2u/db2u.watsonquery@sha256:48cce86af50ba003dbed9ce54b0240b9eec59a2702ab93bae58a649d7fe777ce
  - name: icr.io/db2u/db2u@sha256:0ecd06074f6ce0f090f93b2b5ed51e41679154d638dc2031d64498aab2c0de68
  - name: icr.io/db2u/db2u@sha256:323e043a92970faa921733dbd4d474bab3e80dc105fc6fe4b9c5e0d2160e1856
  - name: icr.io/db2u/db2u@sha256:64d1fc881fa7c4f9203a1e43b498464333768623cc1494d42e2d8aac21b4af3c
  - name: icr.io/db2u/db2u@sha256:9678623a716ceb4ed9367812b842cd56227ed15858edae08f7bcd5961dfa5e2d
  - name: icr.io/db2u/db2u@sha256:b8701f8d719345ee156c2490d8a0fc9ee08a2cfca7c4537d728c729337da9b77
  - name: icr.io/db2u/db2u@sha256:d282a83ba1d016035d9e7c8f489b3599cc3e39826890e927856a5232b6f6013a
  - name: icr.io/db2u/db2u@sha256:d7c92b368064d15ddf6bcc3d96dc19f876603f5a9fb453185724607e2b966567
  - name: icr.io/db2u/db2u@sha256:ef563ca49bf4ad388e8af360a5a1d87fd756d0e9329c6a181f3a37c29429b9c6
  - name: icr.io/db2u/db2u@sha256:f95b4ad43756f1f256df6f9b8cc79757db430de06a0fea22b78261490a73649a
  - name: icr.io/db2u/etcd@sha256:3c5d8ff1c77f146b6db7224bcfc92d6d64a832332cc087a162b2cda81ea99eef
  - name: icr.io/db2u/etcd@sha256:84dda9076ae58f2f02abba6102ab8325e4c301984b2588cb751e46245c636940
  - name: icr.io/db2u/etcd@sha256:9fa13ffeab820c5a62bc5bdb41347ef5cfd5244584117f0f3bd28cfe8690d358
  - name: icr.io/db2u/etcd@sha256:a5123f1a68e08bbda547eb653b724b917c9ec4d4cb6f447b066c4b71c6828909
  - name: icr.io/db2u/etcd@sha256:af1d052b60f599452ecbb44a060173922286e296cbf52d5a551a485cc28900e6
  - name: icr.io/db2u/etcd@sha256:b211cdc9c094ecad0f8bb3d0b714331937bb26582e450b3babec158f9857976a
  - name: icr.io/db2u/etcd@sha256:b617959f55b413b22a5bb72a80c19c7ff966a695dd4793903c9044165296aa3c
  - name: icr.io/db2u/etcd@sha256:cc1f51f48d7d95bd25a1658490adbdc0b7d8d0429e23ff621b3b8b457d51e271
  - name: icr.io/db2u/etcd@sha256:e386ff2ca65b99c63c6166de9a649875891f493091754aa4dc0e83e92dae7cb2
  blockedImages:
  - name: icr.io/cpopen/db2u-day2-operator@sha256:91a7ebbbf0667cf18e6686145dcb9acf2df2d616ee61073309100247e3b7e6c3
  - name: icr.io/cpopen/db2u-operator@sha256:4a938eb64e44e5bf155293e1939c4c8f2bcee9603647167994a94dbd79ced316
  - name: icr.io/cpopen/ibm-db2uoperator-bundle@sha256:8fc35a4bdb7db8231f0258986a3dcad014b80059681a55ce4cd2a87659c52849
  - name: icr.io/cpopen/ibm-db2uoperator-catalog:253.0.0-26025-
hack@sha256:3325c4e4ec118c186f80b8336e5a3a5d8e2c2d7a1f06d97c9fee1cc4c90d68dc
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:12713664b4753551ae3ac303c25c4df938a4394e241e1f9bbc526e74a8ad562e
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:4b890828cc50851a42e5fec6bb929d1f71578aeb143794cbfaa1207eb0d7d048
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:890d04affc2beb381a483162aa3e998599b308ca1615ecd63e66ba04d94c224a
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:b711c164a62ccd4db955ea67afb0e225601429b6406d98fbad31c5fea13a86ea
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:d35b06c1d03707642c6641aff7d481be128064b3aabb384e5aac43715da45d04
  - name: icr.io/db2u/db2u.auxiliary.auth@sha256:e57279502a5d605931494771e8de0b382d3ddb1a5d744820f70977bb777bdaf9
  - name: icr.io/db2u/db2u.db2wh@sha256:1594796ebbe52b158c12eeb8e5d4faa791f6c9a28f96e33844a05f0282b4fc1f
  - name: icr.io/db2u/db2u.db2wh@sha256:394a2a0c640a884c1f63332bce1a8d5ad73baba0d3b3a05b4fa876e35bf9d82e
  - name: icr.io/db2u/db2u.dv.api@sha256:f8ed8a89397c5aa2d6b61ac2ed9d55e3b6351c3c0396355c17d7e44a48038def
  - name: icr.io/db2u/db2u.dv.caching@sha256:d5fe8bbd87c5d4e4424f5bd71ae3bf35769ea744322b152128ab563412eb68b0
  - name: icr.io/db2u/db2u.dv.utils@sha256:83ec4f59d30685a3ab43c74ee20336a16635935f339b78c4744058504e110f5e
  - name: icr.io/db2u/db2u.hurricane@sha256:7e7d7b1df0122120aeb33864b00cd5b9d8a2beab0ee6e5bbe273b0b7ea4f0e14
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:22bb255ac34cff1d99255a52a94b6560880f5f51937fdfcedba1ccd668688d9c
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:61c8a176b449e0aa7638a524297a80ece4ec52700d99299ae855db5b54609438
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:7906196d7910eada7f94bb375f4d6d7e5174992a6a0b3aa3308ce12283df45a8
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:7ea95db7c6249ad9b115c5934b4849f68b633024db937bb7835ec3cb4ab6a897
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:b67acd5ee75833ec1e10705a0c4e7ba0f4be85ab03ed37527b56250ee3448cb9
  - name: icr.io/db2u/db2u.instdb.restricted@sha256:d5e452c1f3f5dcacb747713e56dcd006be6ba291bbf305352cd411a9bae57ee6
  - name: icr.io/db2u/db2u.instdb@sha256:4e8688e4964cbbeb4a5329294a0d995a0a45a9bec237e3f0a18491f0bce97784
  - name: icr.io/db2u/db2u.instdb@sha256:5ad3fc61b531781b212275eed3c3632ba0b7cd7d5cb48cf321b3f2308b66c421
```

```
  - name: icr.io/db2u/db2u.instdb@sha256:829a95722e17914869a4e224f35ace529607e71a1d2235f6e6fe7f33378d1cb5
  - name: icr.io/db2u/db2u.instdb@sha256:a6c008e42a208fe1407cf8e9a9c5e4d97eb923aee7269821fad9bd037b407357
  - name: icr.io/db2u/db2u.instdb@sha256:cc300234190d53a01cbe7349b628df0518553b4cd6795bebfbbacd66ca5ae29b
  - name: icr.io/db2u/db2u.instdb@sha256:d1f0a4cd8591bc509208ab01d0268881d4a8529d53bae147e570de5a0ead3bb1
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:344840dc5a825a755f9701792bfc5f8e7766b776665c88e83a04bec8a1cd85ac
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:7313133d3b83dbf4a214c73e809b21b431bf81f4ff3c027a4ebe0608c254c9fc
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:7e7ac8e3f66e45c99e4fcf8673e09b18e2c0f7a0d6f72c3f86eb2eb157fabaf8
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:88574364d46b83ca98f20569e7ba0533fd38dd6a3cdff23bc374a629de1379e3
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:b8288381e8b2538464097db4328641648a44c5e9671bbc502af2e997b39abb41
  - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:f84b460effa534c060592012f01d85f16d3405579 31e3532f5f6e0b4b892b266
  - name: icr.io/db2u/db2u.mustgather@sha256:51707eef34bad0eeeecfb09c5746782851a0b4d6af562b626dc667ec5e50c723
  - name: icr.io/db2u/db2u.mustgather@sha256:517ae4be20211a68c11d5848f6330ec156bff90c3c64057d273b52d4bb98dae9
  - name: icr.io/db2u/db2u.mustgather@sha256:536bf34b07defd3bc624a7a3973e6d5e701d5fdb949978cfcb1dbd546968c475
  - name: icr.io/db2u/db2u.mustgather@sha256:628ac4c7d5e9b6f527cf451f397da5a8723956a283fa9a3dfe1db2dfc91bc858
  - name: icr.io/db2u/db2u.mustgather@sha256:6c91db906f8dd87bef288595ef2393d910f124c8d31440731af8b260fd2fa55d
  - name: icr.io/db2u/db2u.mustgather@sha256:f0a3a3dd13c7d133d39b25234977a150993334ae45049cb51a7fffc15943cd9a
  - name: icr.io/db2u/db2u.qrep@sha256:3c731b912f43a68b8ab8fef2bd61a4fe5287fac5c251046a160a049730f60f5f
  - name: icr.io/db2u/db2u.qrep@sha256:44724bda2d5336f17bec950e40195f4cb6199291a22a3c6d3d9b47407e0155f1
  - name: icr.io/db2u/db2u.qrep@sha256:9512f41becbcfc6ebf8afcaf35521478fd87e2b2881d856d443b5bb882cb8d79
  - name: icr.io/db2u/db2u.qrep@sha256:d6bcb17bd3ff2776ee3588ddebd946cade42a76f38666ccfc425635b09a6eb12
  - name: icr.io/db2u/db2u.rest@sha256:6e1f90125ad8a4740bf388d16f44e73737379f4dd9b881bc2559aea2d60d9976
  - name: icr.io/db2u/db2u.rest@sha256:778a3bd330aa7c9aa7cf405551856e92d3d8b40d75a509054d8c115e73e7b795
  - name: icr.io/db2u/db2u.rest@sha256:a12723d1671e2321286c22cfad30352c7860cd95f8099e45b4847036ccc0830e
  - name: icr.io/db2u/db2u.rest@sha256:a8158630c0ad0a21592a9e08891035d0d02d7b406c15f4740db77fec125b03ce
  - name: icr.io/db2u/db2u.rest@sha256:bf6576eecd916419d2d521289970094e0c73ded5de5154030678d6543c60a2de
  - name: icr.io/db2u/db2u.rest@sha256:d000f87783bf8963423064de28d069e6877aed8ec94d572c6b22a27313fec8ff
  - name: icr.io/db2u/db2u.restricted@sha256:27e1777bf955d26ce713cb8d6ffc6ee1d8d76c79dcbe94c07d1ec154b12519bd
  - name: icr.io/db2u/db2u.restricted@sha256:33a70ec4786221efe5e5fd397951111fc90ea3b75842ef9558c889034564e91b
  - name: icr.io/db2u/db2u.restricted@sha256:4e4397c45c10f09f723ff11b3b51bb7adb492898d652501abd85511cc660ccc6
  - name: icr.io/db2u/db2u.restricted@sha256:6b204a6026e176bc7c7404fd41ede9399439cf5347698c95b838a78454ace6d9
  - name: icr.io/db2u/db2u.restricted@sha256:9b388eab7b17d543bf9c731a2583bd2606ba6199873ab914a3cdd2065e231385
  - name: icr.io/db2u/db2u.restricted@sha256:c75c667f44a6ab63447d0e72ad2743beb6f34f29f0cdee9b83d83c72ebb2457e
  - name: icr.io/db2u/db2u.tools@sha256:02f1771f4a00db39a5ef741dc39eb84caedfd7393092607994500d9b481b9a54
  - name: icr.io/db2u/db2u.tools@sha256:35ae6c6618b443c49a17ff32f5ba0c098e46fc542482c02e94857eb378ace419
  - name: icr.io/db2u/db2u.tools@sha256:83c657db8954d116a97daae90957ed8b31a89055913de08c92bbe5d30d3c71ec
  - name: icr.io/db2u/db2u.tools@sha256:adeffc101962a8181b4ceba587d046808c6440ca4a516cabdb04f973b5e299e9
  - name: icr.io/db2u/db2u.tools@sha256:e89e56844720d8512e0e0ff65a0331719bc4c203cabf2eb1da329a6dd3d00c70
  - name: icr.io/db2u/db2u.tools@sha256:f22e5bbc28263a51aa45c4a42dfaabee793322594c32a58a13914b977aca300f
  - name: icr.io/db2u/db2u.update.image@sha256:318c6f072f745c191feecb34576bb68681dd9db3644cc32191fba34274332852
  - name: icr.io/db2u/db2u.update.image@sha256:56ec63418fba021fb1eafe54bc6d62106dfa9819a3ce88707fabcadb98b239aa
  - name: icr.io/db2u/db2u.update.image@sha256:5af408ae4cb77592ee6726abe3a281b9a7fac500231d05b45b0a2b05ed88293d
  - name: icr.io/db2u/db2u.update.image@sha256:836a33719793c32744c0f830890633cd1648f282585ae672d3067bd146b1d467
  - name: icr.io/db2u/db2u.update.image@sha256:9a5875da2221ed1e2725a085ec7abd2e0418379686e337987f9496367e7c7a61
  - name: icr.io/db2u/db2u.update.image@sha256:d327bb0e5cc2a453d2fad23b629ba4862b5af9f794fc7f10ecf9d46e5d107f19
  - name: icr.io/db2u/db2u.veleroplugin@sha256:1a2d4657bf14fa60ffa1b0dbca50b696f1ff4ffd28f4c470db67b0ecca5d6fb4
  - name: icr.io/db2u/db2u.veleroplugin@sha256:51114e7afaa01a9002289017a22a2f5c9c0eceeb8f2df857c8148dbb5651e46b
  - name: icr.io/db2u/db2u.veleroplugin@sha256:796518c555504dd1e5b77504ca07397c114ac6753adbf8c8de41dcd2e1b57065
  - name: icr.io/db2u/db2u.veleroplugin@sha256:a2b0b6ff986b618fe880acec97d4af1673f4bffda385e05b6e14301c1b51d472
  - name: icr.io/db2u/db2u.veleroplugin@sha256:b3ecbc09923c288fa8d21d9e36a23a61cbec71a0c43bdb1e8213688bab571626
  - name: icr.io/db2u/db2u.veleroplugin@sha256:f87644162951b7210e450f567dacf718d3480a2f3468b36131b7a43286a6c44f
  - name: icr.io/db2u/db2u.watsonquery@sha256:897237498f224723b5818c50cd37d218032c4a05ebe2bda15bebe99d13ef0521
  - name: icr.io/db2u/db2u@sha256:01bdc550ea7d727fa5715f96c61f7ffe763ee38fc1875132187c40e09d08a6aa
  - name: icr.io/db2u/db2u@sha256:427657ba472b22f66b68d2a4f2ea7564572853083508ae2a8dbf770d1a181b85
  - name: icr.io/db2u/db2u@sha256:4319988ff6a8dff6a97368f253fed5cb268eb0c20de4db9dedd77cc47214f806
  - name: icr.io/db2u/db2u@sha256:6dda7e511b92ffc83c4b92e7b5ed99b5b619a5986af4a86d75c5f4425917dd91
  - name: icr.io/db2u/db2u@sha256:94226fa2a0fa69f8438265f481ffc05e830a75b9bb4a04f2a2b1f8d239758b09
  - name: icr.io/db2u/db2u@sha256:adc37d1fdfde105a516b8b891c494627703d90e9104181b2959a98e6630a4e94
  - name: icr.io/db2u/etcd@sha256:0c79ec0ad02a7e810ce4bc094fe85ad869ac973494161e85dbf32e394fcccf6c
  - name: icr.io/db2u/etcd@sha256:1a3362799bd9402aba7f99eed179c2747a520b37c994e63ad9e2b5fdeb4d5bd9
  - name: icr.io/db2u/etcd@sha256:1befb2b817dc6f7bf3f2fe3f5f2862b638d791d1284facddf789f245775e575c
  - name: icr.io/db2u/etcd@sha256:234c073c178f60ff8e8e29fdfda25b1e908ea0655ff61f3a4e29a51eefdea8eb
  - name: icr.io/db2u/etcd@sha256:6e0c097d8e6d9cdacc71f80312dd96b8a0d50a07ad48a333fce0bfc190f58553
  - name: icr.io/db2u/etcd@sha256:bb5c436b88c57715ffc991d0ab5a05979a4c092c65ad98296af2b1349926c45c
EOF
```

6. Run the following `oc-mirror` command to mirror the images.

```
oc mirror --v2 --config imageset-config-dcs.yaml --workspace file://./ docker://${TARGET_PATH} --dest-tls-verify=false
```

7. Generate `oc-mirror` config files using V2:

```
oc ibm-pak generate mirror-manifests --oc-mirror-plugin=v2 --version "${CASE_VERSION}" "${CASE_NAME}" "${LOCAL_REGISTRY}"
```

8. Important: You can run steps 8 and 9 only when the cluster is up for a freshly installed rack.
   Apply the `ImageDigestMirrorSet` file to your cluster:
   Note: Replace the variable `$TARGET_PATH` with your registry details where images are mirrored.

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: ibm-spectrum-discover
spec:
  imageDigestMirrors:
  - mirrors:
    - ${TARGET_PATH}/cp
    source: cp.icr.io/cp
  - mirrors:
    - ${TARGET_PATH}/cpopen
    source: icr.io/cpopen
  - mirrors:
    - ${TARGET_PATH}/db2u
    source: icr.io/db2u
```

In case of any failure, refer Troubleshooting and known issues in offline mirroring to get the resolution steps.

9. Apply the following `catalogsource` to the cluster:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-spectrum-discover-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-spectrum-discover-253.0.0
  publisher: IBM
  image: ${TARGET_PATH}/cpopen/ibm-spectrum-discover-operator-
catalog@sha256:c8b42eafa039542123c4ba29ef650c4a41996a28fd4504fb7c4e2e62fedab5b1
  sourceType: grpc
```

Mirroring Content-Aware Storage (CAS) images

IBM Fusion supports a range of CAS versions, and it can upgrade independently of IBM Fusion within a certain range. In IBM Fusion 2.13.0, the supported CAS version range is any version < v1.1.0. You can run this procedure multiple times to upgrade to each newer version of CAS within the supported range.

Before you begin

- Before you mirror, go through the prerequisites. See Mirroring prerequisites.
- Use either podman or docker commands to login to the registries. The examples in the procedure used docker; if you are using podman, substitute `podman` for `docker` in the commands.
- For more information about the version support, see IBM Fusion support matrix.

About this task

High level steps to mirror CAS service:

1. Log in to the relevant Docker registries
2. Define variables to configure and mirror the CAS component of IBM Fusion.
3. Run the `oc-mirror` CLI tool to mirror the CAS components.

Note:

- Use this procedure when you want to upgrade your offline CAS installation to a newer version, without having to upgrade the IBM Fusion installation.
- If you choose to mirror specific sub-components, use the same TARGET_PATH variable so that all the components mirror into the same location, and the generated `ImageDigestMirrorSet` remains the same across all components.
- For more information about Air gap setup for network restricted Red Hat® OpenShift® Container Platform clusters, see Offline setup for network restricted Red Hat OpenShift Container Platform clusters.

Procedure

1. Log in to your Docker registry:
   a. Run the following command to login to the Docker registry with your Red Hat enterprise credentials:

      ```
      docker login registry.redhat.io -u <Red Hat enterprise registry username> -p <Red Hat enterprise registry password>
      ```

   b. Log in to the IBM Entitled Container Registry using the IBM entitlement key.

      ```
      docker login cp.icr.io -u cp -p <your entitlement key>
      ```

      Note: Ensure that your entitlement key for IBM Fusion contains the correct entitlement.
   c. Set the following environment variables:

      ```
      export LOCAL_ISF_REGISTRY="<Your enterprise registry host>:<port>"
      export LOCAL_ISF_REPOSITORY="<Your image path>"
      export TARGET_PATH="$LOCAL_ISF_REGISTRY/$LOCAL_ISF_REPOSITORY"
      echo "$TARGET_PATH"
      ```

      Note: Port is a non-mandatory value when setting the `LOCAL_ISF_REGISTRY` variable. You can ignore this if your enterprise registry is accessible and has a secure connection.
      Sample value without port:

      ```
      export LOCAL_ISF_REGISTRY="registryhost.com"
      ```

      ```
      export LOCAL_ISF_REGISTRY="registryhost.com:443"
      export LOCAL_ISF_REPOSITORY="fusion-mirror"
      ```

   d. Run the command to login to the Docker registry with your enterprise registry credentials.

      ```
      docker login $LOCAL_ISF_REGISTRY -u <your enterprise registry username> -p <your enterprise registry password>
      ```

      `LOCAL_ISF_REGISTRY` is your entitlement registry.

      `LOCAL_ISF_REPOSITORY` is the image path in which you want to mirror the images. You can choose your own repository paths. For example, sds-images/isf or sds-images.

2. Create the image set configuration for CAS.

   ```
   cat << EOF > imageset-config-cas.yaml
   kind: ImageSetConfiguration
   apiVersion: mirror.openshift.io/v2alpha1
   ```

```
mirror:
  operators:
  - catalog: icr.io/cpopen/ibm-isf-cas-operator-
catalog@sha256:182acd92bd262539c5cc13470465719cf56398188bc6dfa230ccec1b34d8ecf7
    packages:
    - name: ibm-isf-cas-operator
      channels:
      - name: v1.0
      - name: v1.1
    full: true
  additionalImages:
  - name: icr.io/cpopen/ibm-isf-cas-operator-catalog:1.1.5-
08971@sha256:182acd92bd262539c5cc13470465719cf56398188bc6dfa230ccec1b34d8ecf7
EOF
```

3. Run the following `oc-mirror` command to mirror the images.

```
oc mirror --v2 --config imageset-config-cas.yaml --workspace file://./ docker://${TARGET_PATH} --dest-tls-verify=false
```

4. Apply the `ImageDigestMirrorSet` file to your cluster:
   Note: Replace the variable `$TARGET_PATH` with your registry details where images are mirrored.

```
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: isf-cas
spec:
  imageDigestMirrors:
  - mirrors:
    - ${TARGET_PATH}/cp
    source: cp.icr.io/cp
  - mirrors:
    - ${TARGET_PATH}/cpopen
    source: icr.io/cpopen
  - mirrors:
    - ${TARGET_PATH}/openshift4
    source: registry.redhat.io/openshift4
```

   In case of any failure, refer Troubleshooting and known issues in offline mirroring to get the resolution steps.

   Note: After applying the catalog source, it gets deleted automatically and it recreates once CAS is deployed.
   There is no need to apply the generated catalog-source.yaml file in this directory as Fusion applies the CatalogSource for CAS.

5. Apply the following catalog source to the cluster.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cas-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-cas-1.1.5-linux-amd64
  publisher: IBM
  image: ${TARGET_PATH}/cpopen/ibm-isf-cas-operator-
catalog@sha256:182acd92bd262539c5cc13470465719cf56398188bc6dfa230ccec1b34d8ecf7
  sourceType: grpc
  grpcPodConfig:
    nodeSelector:
      kubernetes.io/arch: amd64
```

Troubleshooting and known issues in offline mirroring

Issues can occur during the offline mirroring of images. This topic outlines known issues and offers troubleshooting guidance.

`oc-mirror` command fails with unauthorized error

Problem statement

The `oc-mirror` command fails with an `unauthorized: access to the requested resource is not authorized` error.

Cause

The error occurs due to one of the following reasons:

- Insufficient authentication credentials for the registries mentioned in the command.
- Incorrect or missing secrets in the docker configuration file.

Resolution

Follow the steps to resolve the issue:

1. Ensure you are logged in to the registries mentioned in the `oc-mirror` command.
2. Verify that the `~/.docker/config.json` file contains the correct secrets for the registries.
   - If the file does not exist, create it manually.
   - Add the required registry credentials to the file.
3. After completing the preceding steps, retry the `oc-mirror` command. If the issue persists, review the registry credentials and docker configuration file for accuracy.

`oc-mirror` v2 fails with an error `stat .: no such file or directory`

Problem statement

The **`oc-mirror`** v2 fails with an error **`stat .: no such file or directory`**.

Resolution

As a resolution, install the **`oc-mirror`** CLI plug-in of **4.20.6** version.

Use the latest versions of the oc and oc-mirror tools

Problem statement

Issues may occur when you do not use the latest version of the **oc** CLI or **oc-mirror** plug in.

Diagnosis

Red Hat® has identified several library compatibility issues in the older versions of these tools, which are resolved in the latest releases. When you run older versions of **oc** or **oc-mirror** commands, you may encounter messages similar to the following examples.

```
oc: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.33' not found (required by oc)
oc: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.34' not found (required by oc)
oc: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.32' not found (required by oc)
```

Resolution

Download the latest available **oc** CLI tool, even if it is a newer than your Red Hat OpenShift® version. The **oc** CLI is backward compatible with older OpenShift versions. For example, your **oc** CLI can be on version 4.17 while your OpenShift cluster remains on 4.14.
In addition, ensure you obtain the latest version of the **oc-mirror** plugin. The **oc-mirror** plugin need not match the **oc** CLI version and can even be newer. For example, you can use an **oc-mirror** plugin 4.17 with an **oc** CLI 4.16, even if your cluster is on 4.14.

Login to both source and target registries before mirroring images

Problem statement

You may encounter issues where some or all images fail to copy, resulting in errors related to either authorization or authentication requirements.

Diagnosis

If you encounter messages similar to the following examples, then it indicates that you must login to the specified registries:

- `error: unable to retrieve source image cp.icr.io/cp/isf/isf-prereq-operator manifest sha256:1d062784dd6818297bd8275d139d0dd7ac6cd2ae6a4188bb3258bcdad8ccedf4: Get "https://cp.icr.io/v2/cp/isf/isf-prereq-operator/manifests/sha256:1d062784dd6818297bd8275d139d0dd7ac6cd2ae6a4188bb325 unauthorized: Authorization required. See https://cloud.ibm.com/docs/Registry?topic=Registry-troubleshoot-auth-req`

- `Checking push permissions for docker-na-public.artifactory.swg-devops.com error: error checking push permissions for docker-na-public.artifactory.swg-devops.com: creating push check transport for docker-na-public.artifactory.swg-devops.com failed: GET https://docker-na-public.artifactory.swg-devops.com/artifactory/api/docker/null/v2/token?scope=repository%3Aoffline%2 Authentication is required`

Resolution

As a resolution, log in to all the registries:

- registry.redhat.io
- icr.io
- quay.io
- Your target registry host

Also, create a **`.docker/config.json`** file in MGen with

- registry.redhat.io
- icr.io
- quay.io
- Your target registry host

If authorization errors persist even after logging, it indicates that your user does not have the right permissions to pull from the source registries or push to target registries. In such a case, contact your registry administrator regarding your user permissions

Check available disk space

Problem statement

If repeated retries of **oc-mirror** continue to fail, it is possible that the target registry has run out of disk space. Additionally, if you mirror to a local directory for offline install media, your local disk may be full.

Resolution

Check disk space usage and free up disk space if necessary.

Error with HTTP status: 504 Gateway Timeout or no space left on device

Problem statement

The **oc-mirror** operation may fail with an error message similar to the following example:

```
Rendering catalog image
"our.offline-registry.net/fusion-hci-2.9.0/cpopen/isf-operator-catalog:f34ee5" with file-based catalog
error: error rebuilding catalog images from file-based catalogs:
error regenerating the cache for our.offline-registry.net /fusion-hci-2.9.0/cpopen/isf-operator-catalog:f34ee5: exit status 1
```

Resolution

This is a known issue in the **oc-mirror** tool. For more information about how to resolve the issue, see the Red Hat article.

Error with HTTP status: 502 Bad Gateway

Problem statement

The **oc-mirror** operation may fail with an error message similar to the following example:

```
error: unable to retrieve source image registry.redhat.io/amq-streams/kafka-34-rhel8 manifest sha256:e4b2073b3001502b3bd02275e
error: unable to retrieve source image registry.access.redhat.com/ubi8/openjdk-11 manifest sha256:d9739b266415be7c915daf3c61f8
error: an error occurred during planning
```

Resolution

As a resolution, retry the **oc-mirror** command.

Error with HTTP status: 504 Gateway Timeout or no space left on the device

Problem statement

The **oc-mirror** operation may fail with an error message similar to the following example:

```
error: unable to retrieve source image registry.redhat.io/amq-streams/kafka-35-rhel8 manifest #3 from manifest list: received
error: unable to retrieve source image registry.redhat.io/amq-streams/kafka-35-rhel8 manifest #1 from manifest list: received
error: unable to retrieve source image registry.redhat.io/amq-streams/kafka-35-rhel8 manifest #1 from manifest list: received
error: an error occurred during planning
```

Resolution

1. Run the following command:

   ```
   rm -rf /root/.oc-mirror
   ```

2. Retry the **oc-mirror** command.

Installing IBM Fusion on On-premises VMware

Procedure to install IBM Fusion on OpenShift® Container Platform that runs on On-premises VMware.

Before you begin

- If you plan to do an offline installation of IBM Fusion, see Enterprise registry for IBM Fusion installation.
- Ensure you complete all the prerequisites before you proceed with the installation. For the prerequisites, see Prerequisites.
- For more information about deployments and their supported services, see Support matrix for IBM Fusion versions.

About this task

Support is available for only one instance of IBM Fusion per OpenShift Container Platform.

Procedure

1. If you have not configured the IBM Operator Catalog, then configure it. For the procedure to add IBM Operator catalog, see Adding the IBM operator catalog.
   Note:
   - IBM Operator Catalog is not specific to a version of IBM Fusion.
   - If you already have the IBM Maximo® software configured in your environment or have plans to configure it, use only the **ibm-operator-catalog** of IBM Maximo® software.
2. Log in to Red Hat® OpenShift Container Platform web management console.
3. Go to Operators > OperatorHub.
4. Under Source, select IBM Operator Catalog.
   It lists all operators that are part of the IBM Operator Catalog including **IBM Fusion**.
5. Click IBM Fusion.
   The Version, Capability level, Source, and Provider type of IBM Fusion is available.
6. Click Install.
   It opens the Install Operator page for **IBM Fusion** operator.
7. Select `v2.0` in the update channel where the current operator is published.
   Note: You can also subscribe for updates. The subscription to the channel helps to keep the operator up to date.
8. In the Installation mode, select A specific namespace on the cluster.
   The operator will be available in a single Namespace only.
9. Select **ibm-spectrum-fusion-ns** in the Operator recommended namespace. Alternatively, use the Select a namespace option to select an existing namespace or create a new namespace.
10. Important: Always set Update approval to Manual as IBM Fusion does not support Automatic.
    In the Update approval section, you can select either Manual or Automatic strategy.

Always set Update approval to Manual because the Automatic option automatically upgrades the operator whenever a new version of the operator is released to the channel. This automatic upgrade might have an impact on your running workloads.

11. Click Install.
    The installation of the operator begins.
12. Wait for the operator to complete the installation.
    After the successful installation of the operator, the following message gets displayed:

    ```
    Installed operator - operand required
    ```

    Generally, it completes in few minutes. If it takes more time, check whether all pods are up and running.

    Note: If you notice that installation is taking longer than usual or failed, then check the events tab on the **installed-opertor** or **fusion-operator** page. In the events tab, if you see an error **waiting for deployment of serviceability-operator**, ignore it and continue with the installation.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Fusion.
    The License agreement page gets displayed.
14. Go through the license agreement, click I have read and accept the license agreement, and click Continue.
    The Welcome to IBM Fusion dialog box gets displayed.
15. In the IBM Fusion user interface, click Install services to install the services right away or click Maybe later to do it later.

What to do next

1. For steps to verify the success of the installation, see Validating IBM Fusion installation.
2. If you want to enable the services from the IBM Fusion user interface, see the following procedures accordingly:
   a. For IBM Data Cataloging, see Installing IBM Data Cataloging.
   b. For Data Foundation, see Installing Fusion Data Foundation.
   c. For Backup & Restore, see Install Backup & Restore.
   d. For CAS, see Installing Content-Aware Storage (CAS).

   If you want to install IBM Fusion services using OpenShift Container Platform, then see Deploying IBM Fusion services using CLI.
3. For more information about the user interface of IBM Fusion, see Knowing your IBM Fusion user interface.

Installing IBM Fusion on On-premises Bare Metal

Procedure to install IBM Fusion on OpenShift® Container Platform that runs on On-premises Bare Metal.

Before you begin

- If you plan to do an offline installation of IBM Fusion, see Enterprise registry for IBM Fusion installation.
- For more information about deployments and their supported services, see Support matrix for IBM Fusion versions.

Ensure you complete all the prerequisites before you proceed with the installation. For the prerequisites, see Prerequisites.

About this task

Support is available for only one instance of IBM Fusion per OpenShift Container Platform.

Procedure

1. If you have not configured the IBM Operator Catalog, then configure it. For the procedure to add IBM Operator catalog, see Adding the IBM operator catalog.
   Note:
   - IBM Operator Catalog is not specific to a version of IBM Fusion.
   - If you already have the IBM Maximo® software configured in your environment or have plans to configure it, use only the **ibm-operator-catalog** of IBM Maximo® software.
2. Log in to Red Hat® OpenShift Container Platform web management console.
3. Go to Operators > OperatorHub.
4. Under Source, select IBM Operator Catalog.
   It lists all operators that are part of the IBM Operator Catalog including **IBM Fusion**.
5. Click IBM Fusion.
   The Version, Capability level, Source, and Provider type of IBM Fusion is available.
6. Click Install.
   It opens the Install Operator page for **IBM Fusion** operator.
7. Select `v2.0` in the update channel where the current operator is published.
   Note: You can also subscribe for updates. The subscription to the channel helps to keep the operator up to date.
8. In the Installation mode, select A specific namespace on the cluster.
   The operator will be available in a single Namespace only.
9. Select **ibm-spectrum-fusion-ns** in the Operator recommended namespace. Alternatively, use the Select a namespace option to select an existing namespace or create a new namespace.
10. Important: Always set Update approval to Manual as IBM Fusion does not support Automatic.
    In the Update approval section, you can select either Manual or Automatic strategy.
    Always set Update approval to Manual because the Automatic option automatically upgrades the operator whenever a new version of the operator is released to the channel. This automatic upgrade might have an impact on your running workloads.
11. Click Install.
    The installation of the operator begins.
12. Wait for the operator to complete the installation.
    After the successful installation of the operator, the following message gets displayed:

`Installed operator - operand required`

Generally, it completes in few minutes. If it takes more time, check whether all pods are up and running.

Note: If you notice that installation is taking longer than usual or failed, then check the events tab on the **`installed-opertor`** or **`fusion-operator`** page. In the events tab, if you see an error **`waiting for deployment of serviceability-operator`**, ignore it and continue with the installation.

13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Fusion.
    The License agreement page gets displayed.
14. Go through the license agreement, click I have read and accept the license agreement, and click Continue.
    The Welcome to IBM Fusion dialog box gets displayed.
15. In the IBM Fusion user interface, click Install services to install the services right away or click Maybe later to do it later.

What to do next

1. For steps to verify the success of the installation, see Validating IBM Fusion installation.
2. If you want to enable the services from the IBM Fusion user interface, see the following procedures accordingly:
   a. For IBM Data Cataloging, see Installing IBM Data Cataloging.
   b. For Data Foundation, see Installing Fusion Data Foundation.
   c. For Backup & Restore, see Install Backup & Restore.
   d. For CAS, see Installing Content-Aware Storage (CAS).
   If you want to install IBM Fusion services using OpenShift Container Platform, then see Deploying IBM Fusion services using CLI.
3. For more information about the user interface of IBM Fusion, see Knowing your IBM Fusion user interface.

Installing IBM Fusion on On-premises Linux on IBM Z and zCX

Procedure to install IBM Fusion on OpenShift® Container Platform that runs on On-premises Linux on IBM Z and zCX.

Before you begin

If you plan to do an offline installation of IBM Fusion, see Enterprise registry for IBM Fusion installation.

Ensure you complete all the prerequisites before you proceed with the installation. For the prerequisites, see Prerequisites.

For more information about deployments and their supported services, see Support matrix for IBM Fusion versions.

About this task

Support is available for only one instance of IBM Fusion per OpenShift Container Platform.

For reference architecture document that describes the use of IBM Fusion or Data Foundation on IBM LinuxONE, see IBM Fusion and Fusion Data Foundation Reference Architecture.

Procedure

1. If you have not configured the IBM Operator Catalog, then configure it. For the procedure to add IBM Operator catalog, see Adding the IBM operator catalog.
   Note:
   - IBM Operator Catalog is not specific to a version of IBM Fusion.
   - If you already have the IBM Maximo® software configured in your environment or have plans to configure it, use only the **`ibm-operator-catalog`** of IBM Maximo® software.
2. Log in to Red Hat® OpenShift Container Platform web management console.
3. Go to Operators > OperatorHub.
4. Under Source, select IBM Operator Catalog.
   It lists all operators that are part of the IBM Operator Catalog including **`IBM Fusion`**.
5. Click IBM Fusion.
   The Version, Capability level, Source, and Provider type of IBM Fusion is available.
6. Click Install.
   It opens the Install Operator page for **`IBM Fusion`** operator.
7. Select `v2.0` in the update channel where the current operator is published.
   Note: You can also subscribe for updates. The subscription to the channel helps to keep the operator up to date.
8. In the Installation mode, select A specific namespace on the cluster.
   The operator will be available in a single Namespace only.
9. Select **`ibm-spectrum-fusion-ns`** in the Operator recommended namespace. Alternatively, use the Select a namespace option to select an existing namespace or create a new namespace.
10. Important: Always set Update approval to Manual as IBM Fusion does not support Automatic.
    In the Update approval section, you can select either Manual or Automatic strategy.
    Always set Update approval to Manual because the Automatic option automatically upgrades the operator whenever a new version of the operator is released to the channel. This automatic upgrade might have an impact on your running workloads.
11. Click Install.
    The installation of the operator begins.
12. Wait for the operator to complete the installation.
    After the successful installation of the operator, the following message gets displayed:

    `Installed operator - operand required`

    Generally, it completes in few minutes. If it takes more time, check whether all pods are up and running.

Note: If you notice that installation is taking longer than usual or failed, then check the events tab on the **installed-opertor** or **fusion-operator** page. In the events tab, if you see an error **waiting for deployment of serviceability-operator**, ignore it and continue with the installation.

13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Fusion.
    The License agreement page gets displayed.
14. Go through the license agreement, click I have read and accept the license agreement, and click Continue.
    The Welcome to IBM Fusion dialog box gets displayed.
15. In the IBM Fusion user interface, click Install services to install the services right away or click Maybe later to do it later.

What to do next

1. For steps to verify the success of the installation, see Validating IBM Fusion installation.
2. If you want to enable the services from the IBM Fusion user interface, see the following procedures accordingly:
    a. For IBM Data Cataloging, see Installing IBM Data Cataloging.
    b. For Data Foundation, see Installing Fusion Data Foundation.
    c. For Backup & Restore, see Install Backup & Restore.
    d. For CAS, see Installing Content-Aware Storage (CAS).
    If you want to install IBM Fusion services using OpenShift Container Platform, then see Deploying IBM Fusion services using CLI.
3. For more information about the user interface of IBM Fusion, see Knowing your IBM Fusion user interface.

Installing IBM Fusion on IBM Cloud

Procedure to install IBM Fusion on OpenShift® Container Platform that runs on IBM Cloud®.

Before you begin

Ensure you complete all the prerequisites before you proceed with the installation. For the prerequisites, see Prerequisites.

For more information about deployments and their supported services, see Support matrix for IBM Fusion versions.

For steps to create a image pull secret for installing IBM Fusion on IBM Cloud, see Creating image pull secret for IBM Cloud based installation.

About this task

Support is available for only one instance of IBM Fusion per OpenShift Container Platform.

Procedure

1. If you have not configured the IBM Operator Catalog, then configure it. For the procedure to add IBM Operator catalog, see Adding the IBM operator catalog.
    Note:
    - IBM Operator Catalog is not specific to a version of IBM Fusion.
    - If you already have the IBM Maximo® software configured in your environment or have plans to configure it, use only the **ibm-operator-catalog** of IBM Maximo® software.
2. Log in to Red Hat OpenShift Container Platform web management console.
3. Go to Operators > OperatorHub.
4. Under Source, select IBM Operator Catalog.
    It lists all operators that are part of the IBM Operator Catalog including **IBM Fusion**.
5. Click IBM Fusion.
    The Version, Capability level, Source, and Provider type of IBM Fusion is available.
6. Click Install.
    It opens the Install Operator page for **IBM Fusion** operator.
7. Select `v2.0` in the update channel where the current operator is published.
    Note: You can also subscribe for updates. The subscription to the channel helps to keep the operator up to date.
8. In the Installation mode, select A specific namespace on the cluster.
    The operator will be available in a single Namespace only.
9. Select **ibm-spectrum-fusion-ns** in the Operator recommended namespace. Alternatively, use the Select a namespace option to select an existing namespace or create a new namespace.
10. Important: Always set Update approval to Manual as IBM Fusion does not support Automatic.
    In the Update approval section, you can select either Manual or Automatic strategy.
    Always set Update approval to Manual because the Automatic option automatically upgrades the operator whenever a new version of the operator is released to the channel. This automatic upgrade might have an impact on your running workloads.
11. Click Install.
    The installation of the operator begins.
12. Wait for the operator to complete the installation.
    After the successful installation of the operator, the following message gets displayed:

    ```
    Installed operator - operand required
    ```

    Generally, it completes in few minutes. If it takes more time, check whether all pods are up and running.

    Note: If you notice that installation is taking longer than usual or failed, then check the events tab on the **installed-opertor** or **fusion-operator** page. In the events tab, if you see an error **waiting for deployment of serviceability-operator**, ignore it and continue with the installation.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Fusion.
    The License agreement page gets displayed.
14. Go through the license agreement, click I have read and accept the license agreement, and click Continue.
    The Welcome to IBM Fusion dialog box gets displayed.

15. In the IBM Fusion user interface, click Install services to install the services right away or click Maybe later to do it later.

What to do next

1. For steps to verify the success of the installation, see Validating IBM Fusion installation.
2. If you want to enable the services from the IBM Fusion user interface, see the following procedures accordingly:
   a. For IBM Data Cataloging, see Installing IBM Data Cataloging.
   b. For Data Foundation, see Installing Fusion Data Foundation.
   c. For Backup & Restore, see Install Backup & Restore.
   d. For CAS, see Installing Content-Aware Storage (CAS).

   If you want to install IBM Fusion services using OpenShift Container Platform, then see Deploying IBM Fusion services using CLI.
3. For more information about the user interface of IBM Fusion, see Knowing your IBM Fusion user interface.

Installing IBM Fusion on Amazon Web Services

Procedure to install IBM Fusion on OpenShift® Container Platform that runs on Amazon Web Services and Amazon Web Services ROSA.

Before you begin

- Ensure you complete all the prerequisites before you proceed with the installation. For the prerequisites, see Prerequisites
- For more information about deployments and their supported services, see Support matrix for IBM Fusion versions.
- Configure image pull secrets. For more information about pull secret prerequisite, see Creating image pull secret.
- Multiple zone Amazon Web Services cluster
- Open TCP ports on Amazon Web Services: 1191, 12345, 60000 –61000. Sample script to open these ports:

```
ROSA_NAME=YOUR_ROSA_CLUSTER_NAMECLUSTER_REGION="$(rosa describe cluster -c $ROSA_NAME| grep Region | awk '{print $2}')"VPC_ID="$(aws ec2 describe-security-groups --region ${CLUSTER_REGION} | grep -A15 $ ROSA_NAME | grep VpcId | tail -1 | cut -d '"' -f4)"SG_ID="$(aws ec2 describe-security-groups --region ${CLUSTER_REGION} --filters "Name=vpc-id,Values=${VPC_ID}" "Name=tag:Name,Values=*-worker-sg" --output json --query "SecurityGroups[*].[GroupId]" | jq | grep sg | awk -F '"' '{print $2}')"aws ec2 authorize-security-group-ingress --region ${CLUSTER_REGION} --group-id "${SG_ID}" --protocol tcp --port 12345 --source-group "${SG_ID}" | jq | tee -a ${SETUP_LOG}aws ec2 authorize-security-group-ingress --region ${CLUSTER_REGION} --group-id "${SG_ID}" --protocol tcp --port 1191 --source-group "${SG_ID}" | jq | tee -a ${SETUP_LOG}aws ec2 authorize-security-group-ingress --region ${CLUSTER_REGION} --group-id "${SG_ID}" --protocol tcp --port "60000-61000" --source-group "${SG_ID}" | jq | tee -a ${SETUP_LOG}
```

About this task

Support is available for only one instance of IBM Fusion per OpenShift Container Platform.

Procedure

1. If you have not configured the IBM Operator Catalog, then configure it. For the procedure to add IBM Operator catalog, see Adding the IBM operator catalog.
   Note:
   - IBM Operator Catalog is not specific to a version of IBM Fusion.
   - If you already have the IBM Maximo® software configured in your environment or have plans to configure it, use only the **ibm-operator-catalog** of IBM Maximo® software.
2. Log in to Red Hat® OpenShift Container Platform web management console.
3. Go to Operators > OperatorHub.
4. Under Source, select IBM Operator Catalog.
   It lists all operators that are part of the IBM Operator Catalog including **IBM Fusion**.
5. Click IBM Fusion.
   The Version, Capability level, Source, and Provider type of IBM Fusion is available.
6. Click Install.
   It opens the Install Operator page for **IBM Fusion** operator.
7. Select `v2.0` in the update channel where the current operator is published.
   Note: You can also subscribe for updates. The subscription to the channel helps to keep the operator up to date.
8. In the Installation mode, select A specific namespace on the cluster.
   The operator will be available in a single Namespace only.
9. Select **ibm-spectrum-fusion-ns** in the Operator recommended namespace. Alternatively, use the Select a namespace option to select an existing namespace or create a new namespace.
10. Important: Always set Update approval to Manual as IBM Fusion does not support Automatic.
    In the Update approval section, you can select either Manual or Automatic strategy.
    Always set Update approval to Manual because the Automatic option automatically upgrades the operator whenever a new version of the operator is released to the channel. This automatic upgrade might have an impact on your running workloads.
11. Click Install.
    The installation of the operator begins.
12. Wait for the operator to complete the installation.
    After the successful installation of the operator, the following message gets displayed:

    ```
    Installed operator - operand required
    ```

    Generally, it completes in few minutes. If it takes more time, check whether all pods are up and running.

    Note: If you notice that installation is taking longer than usual or failed, then check the events tab on the **installed-opertor** or **fusion-operator** page. In the events tab, if you see an error **waiting for deployment of serviceability-operator**, ignore it and continue with the installation.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Fusion.

The License agreement page gets displayed.
14. Go through the license agreement, click I have read and accept the license agreement, and click Continue.
The Welcome to IBM Fusion dialog box gets displayed.
15. In the IBM Fusion user interface, click Install services to install the services right away or click Maybe later to do it later.

What to do next

1. For steps to verify the success of the installation, see Validating IBM Fusion installation.
2. If you want to enable the services from the IBM Fusion user interface, see the following procedures accordingly:
    a. For IBM Data Cataloging, see Installing IBM Data Cataloging.
    b. For Data Foundation, see Installing Fusion Data Foundation.
    c. For Backup & Restore, see Install Backup & Restore.
    d. For CAS, see Installing Content-Aware Storage (CAS).
    If you want to install IBM Fusion services using OpenShift Container Platform, then see Deploying IBM Fusion services using CLI.
3. For more information about the user interface of IBM Fusion, see Knowing your IBM Fusion user interface.

Installing IBM Fusion on Microsoft Azure

Procedure to install IBM Fusion on OpenShift® Container Platform that runs on Microsoft Azure.

Before you begin

Ensure you complete all the prerequisites before you proceed with the installation. For the prerequisites, see Prerequisites.

For more information about deployments and their supported services, see Support matrix for IBM Fusion versions.

About this task

Support is available for only one instance of IBM Fusion per OpenShift Container Platform.

Procedure

1. If you have not configured the IBM Operator Catalog, then configure it. For the procedure to add IBM Operator catalog, see Adding the IBM operator catalog.
Note:
    - IBM Operator Catalog is not specific to a version of IBM Fusion.
    - If you already have the IBM Maximo® software configured in your environment or have plans to configure it, use only the **`ibm-operator-catalog`** of IBM Maximo® software.
2. Log in to Red Hat® OpenShift Container Platform web management console.
3. Go to Operators > OperatorHub.
4. Under Source, select IBM Operator Catalog.
It lists all operators that are part of the IBM Operator Catalog including **`IBM Fusion`**.
5. Click IBM Fusion.
The Version, Capability level, Source, and Provider type of IBM Fusion is available.
6. Click Install.
It opens the Install Operator page for **`IBM Fusion`** operator.
7. Select `v2.0` in the update channel where the current operator is published.
Note: You can also subscribe for updates. The subscription to the channel helps to keep the operator up to date.
8. In the Installation mode, select A specific namespace on the cluster.
The operator will be available in a single Namespace only.
9. Select **`ibm-spectrum-fusion-ns`** in the Operator recommended namespace. Alternatively, use the Select a namespace option to select an existing namespace or create a new namespace.
10. Important: Always set Update approval to Manual as IBM Fusion does not support Automatic.
In the Update approval section, you can select either Manual or Automatic strategy.
Always set Update approval to Manual because the Automatic option automatically upgrades the operator whenever a new version of the operator is released to the channel. This automatic upgrade might have an impact on your running workloads.
11. Click Install.
The installation of the operator begins.
12. Wait for the operator to complete the installation.
After the successful installation of the operator, the following message gets displayed:

    ```
    Installed operator - operand required
    ```

    Generally, it completes in few minutes. If it takes more time, check whether all pods are up and running.

    Note: If you notice that installation is taking longer than usual or failed, then check the events tab on the **`installed-opertor`** or **`fusion-operator`** page. In the events tab, if you see an error **`waiting for deployment of serviceability-operator`**, ignore it and continue with the installation.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Fusion.
The License agreement page gets displayed.
14. Go through the license agreement, click I have read and accept the license agreement, and click Continue.
The Welcome to IBM Fusion dialog box gets displayed.
15. In the IBM Fusion user interface, click Install services to install the services right away or click Maybe later to do it later.

What to do next

1. For steps to verify the success of the installation, see Validating IBM Fusion installation.
2. If you want to enable the services from the IBM Fusion user interface, see the following procedures accordingly:
    a. For IBM Data Cataloging, see Installing IBM Data Cataloging.
    b. For Data Foundation, see Installing Fusion Data Foundation.
    c. For Backup & Restore, see Install Backup & Restore.
    d. For CAS, see Installing Content-Aware Storage (CAS).

    If you want to install IBM Fusion services using OpenShift Container Platform, then see Deploying IBM Fusion services using CLI.
3. For more information about the user interface of IBM Fusion, see Knowing your IBM Fusion user interface.

Installing IBM Fusion on On-premises IBM Power Systems

Procedure to install IBM Fusion on OpenShift® Container Platform that runs on On-premises IBM Power Systems.

Before you begin

- If you plan to do an offline installation of IBM Fusion, see Enterprise registry for IBM Fusion installation.
- Ensure you complete all the prerequisites before you proceed with the installation. For the prerequisites, see Prerequisites.
- For more information about deployments and their supported services, see Support matrix for IBM Fusion versions.

About this task

Support is available for only one instance of IBM Fusion per OpenShift Container Platform.

Procedure

1. If you have not configured the IBM Operator Catalog, then configure it. For the procedure to add IBM Operator catalog, see Adding the IBM operator catalog.
   Note:
   - IBM Operator Catalog is not specific to a version of IBM Fusion.
   - If you already have the IBM Maximo® software configured in your environment or have plans to configure it, use only the **`ibm-operator-catalog`** of IBM Maximo® software.
2. Log in to Red Hat® OpenShift Container Platform web management console.
3. Go to Operators > OperatorHub.
4. Under Source, select IBM Operator Catalog.
   It lists all operators that are part of the IBM Operator Catalog including **`IBM Fusion`**.
5. Click IBM Fusion.
   The Version, Capability level, Source, and Provider type of IBM Fusion is available.
6. Click Install.
   It opens the Install Operator page for **`IBM Fusion`** operator.
7. Select `v2.0` in the update channel where the current operator is published.
   Note: You can also subscribe for updates. The subscription to the channel helps to keep the operator up to date.
8. In the Installation mode, select A specific namespace on the cluster.
   The operator will be available in a single Namespace only.
9. Select **`ibm-spectrum-fusion-ns`** in the Operator recommended namespace. Alternatively, use the Select a namespace option to select an existing namespace or create a new namespace.
10. Important: Always set Update approval to Manual as IBM Fusion does not support Automatic.
    In the Update approval section, you can select either Manual or Automatic strategy.
    Always set Update approval to Manual because the Automatic option automatically upgrades the operator whenever a new version of the operator is released to the channel. This automatic upgrade might have an impact on your running workloads.
11. Click Install.
    The installation of the operator begins.
12. Wait for the operator to complete the installation.
    After the successful installation of the operator, the following message gets displayed:

    ```
    Installed operator - operand required
    ```

    Generally, it completes in few minutes. If it takes more time, check whether all pods are up and running.

    Note: If you notice that installation is taking longer than usual or failed, then check the events tab on the **`installed-opertor`** or **`fusion-operator`** page. In the events tab, if you see an error **`waiting for deployment of serviceability-operator`**, ignore it and continue with the installation.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Fusion.
    The License agreement page gets displayed.
14. Go through the license agreement, click I have read and accept the license agreement, and click Continue.
    The Welcome to IBM Fusion dialog box gets displayed.
15. In the IBM Fusion user interface, click Install services to install the services right away or click Maybe later to do it later.

What to do next

1. For steps to verify the success of the installation, see Validating IBM Fusion installation.
2. If you want to enable the services from the IBM Fusion user interface, see the following procedures accordingly:
    a. For IBM Data Cataloging, see Installing IBM Data Cataloging.
    b. For Data Foundation, see Installing Fusion Data Foundation.
    c. For Backup & Restore, see Install Backup & Restore.
    d. For CAS, see Installing Content-Aware Storage (CAS).

    If you want to install IBM Fusion services using OpenShift Container Platform, then see Deploying IBM Fusion services using CLI.

3. For more information about the user interface of IBM Fusion, see Knowing your IBM Fusion user interface.

Installing IBM Fusion on Google Cloud

Procedure to install IBM Fusion on OpenShift® Container Platform that runs on Google Cloud.

Before you begin

- If you plan to do an offline installation of IBM Fusion, see Enterprise registry for IBM Fusion installation.
- For more information about deployments and their supported services, see Support matrix for IBM Fusion versions.

Ensure you complete all the prerequisites before you proceed with the installation. For the prerequisites, see Prerequisites.

About this task

Support is available for only one instance of IBM Fusion per OpenShift Container Platform.

Procedure

1. If you have not configured the IBM Operator Catalog, then configure it. For the procedure to add IBM Operator catalog, see Adding the IBM operator catalog.
   Note:
   - IBM Operator Catalog is not specific to a version of IBM Fusion.
   - If you already have the IBM Maximo® software configured in your environment or have plans to configure it, use only the **`ibm-operator-catalog`** of IBM Maximo® software.
2. Log in to Red Hat® OpenShift Container Platform web management console.
3. Go to Operators > OperatorHub.
4. Under Source, select IBM Operator Catalog.
   It lists all operators that are part of the IBM Operator Catalog including **`IBM Fusion`**.
5. Click IBM Fusion.
   The Version, Capability level, Source, and Provider type of IBM Fusion is available.
6. Click Install.
   It opens the Install Operator page for **`IBM Fusion`** operator.
7. Select `v2.0` in the update channel where the current operator is published.
   Note: You can also subscribe for updates. The subscription to the channel helps to keep the operator up to date.
8. In the Installation mode, select A specific namespace on the cluster.
   The operator will be available in a single Namespace only.
9. Select **`ibm-spectrum-fusion-ns`** in the Operator recommended namespace. Alternatively, use the Select a namespace option to select an existing namespace or create a new namespace.
10. Important: Always set Update approval to Manual as IBM Fusion does not support Automatic.
    In the Update approval section, you can select either Manual or Automatic strategy.
    Always set Update approval to Manual because the Automatic option automatically upgrades the operator whenever a new version of the operator is released to the channel. This automatic upgrade might have an impact on your running workloads.
11. Click Install.
    The installation of the operator begins.
12. Wait for the operator to complete the installation.
    After the successful installation of the operator, the following message gets displayed:

    ```
    Installed operator - operand required
    ```

    Generally, it completes in few minutes. If it takes more time, check whether all pods are up and running.

    Note: If you notice that installation is taking longer than usual or failed, then check the events tab on the **`installed-opertor`** or **`fusion-operator`** page. In the events tab, if you see an error **`waiting for deployment of serviceability-operator`**, ignore it and continue with the installation.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Fusion.
    The License agreement page gets displayed.
14. Go through the license agreement, click I have read and accept the license agreement, and click Continue.
    The Welcome to IBM Fusion dialog box gets displayed.
15. In the IBM Fusion user interface, click Install services to install the services right away or click Maybe later to do it later.

What to do next

1. For steps to verify the success of the installation, see Validating IBM Fusion installation.
2. If you want to enable the services from the IBM Fusion user interface, see the following procedures accordingly:
   a. For IBM Data Cataloging, see Installing IBM Data Cataloging.
   b. For Data Foundation, see Installing Fusion Data Foundation.
   c. For Backup & Restore, see Install Backup & Restore.
   d. For CAS, see Installing Content-Aware Storage (CAS).
   If you want to install IBM Fusion services using OpenShift Container Platform, then see Deploying IBM Fusion services using CLI.
3. For more information about the user interface of IBM Fusion, see Knowing your IBM Fusion user interface.

Installing IBM Fusion from enterprise registry

Prerequisites and procedure to install IBM Fusion on OpenShift® Container Platform.

- **Installing IBM Fusion On-premises**
  Procedure to install IBM Fusion on OpenShift Container Platform.

Installing IBM Fusion On-premises

Procedure to install IBM Fusion on OpenShift® Container Platform.

Before you begin

- Ensure you complete all the prerequisites before you proceed with the installation. For the prerequisites, see Prerequisites.
- If you plan to do an offline installation of IBM Fusion, mirror IBM Fusion, IBM Storage Scale, Data Foundation, Backup & Restore, and IBM® Spectrum Discover image repositories to your registry. For the actual steps, see Enterprise registry for IBM Fusion installation.
- To enable Fusion Data Foundation deployed on OpenShift Container Platform, ensure a `catalogsource` named `redhat-operators` exists in `openshift-marketplace` with Fusion Data Foundation and LocalStorage packages and their dependency packages.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
 name: redhat-operators
 namespace: openshift-marketplace
spec:
 displayName: Red Hat Operators
 image: <enterprise registry host:port>/<target-path>/redhat/redhat-operator-index:v<OCP_VERSION>
 publisher: Red Hat
 sourceType: grpc
```

About this task

Support is available for only one instance of IBM Fusion per OpenShift Container Platform.

Procedure

1. Log in to Red Hat® OpenShift Container Platform web management console.
2. Edit global pull-secret specifying enterprise registry credentials.
3. Add the IBM Fusion operator catalog.

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: isf-catalog
     namespace: openshift-marketplace
   spec:
     displayName: ISF Catalog
     image: <isf catalog image available in your enterprise registry>
     publisher: IBM
     sourceType: grpc
     updateStrategy:
       registryPoll:
         interval: 30m0s
   ```

   See the following sample value:

   ```
   image: registryhost.com:443/fusion-mirror/isf-operator-software-catalog:2.8.0
   ```

4. Go to Operators > OperatorHub.
5. Look for ISF Catalog under OperatorHub.
6. In the Installation mode, select A specific namespace on the cluster.
   The operator will be available in a single Namespace only.
7. Select `ibm-spectrum-fusion-ns` in the Operator recommended namespace. Alternatively, use the Select a namespace option to select an existing namespace or create a new namespace.
8. Important: Always set Update approval to Manual as IBM Fusion does not support Automatic.
   In the Update approval section, you can select either Manual or Automatic strategy.
   Always set Update approval to Manual because the Automatic option automatically upgrades the operator whenever a new version of the operator is released to the channel. This automatic upgrade might have an impact on your running workloads.
9. Click Install.
   The installation of the operator begins.
10. Wait for the operator to complete the installation.
    After the successful installation of the operator, the following message gets displayed:

    ```
    Installed operator - succeeded
    ```

    Generally, it completes in few minutes. If it takes more time, check whether all pods are up and running.

11. Open the IBM Fusion user interface. In the OpenShift Container Platform web console, click the Applications icon in the title bar and select IBM Fusion.
    The License agreement page gets displayed.
12. Go through the license agreement and click I have read and accept the license agreement.
    If you want to download the license agreement copy to your local, click Download a copy.
    The Welcome to IBM Fusion dialog box gets displayed.
13. Click Install services to install the services right away or click May be later to do it later. For the procedure to enable from IBM Fusion, see IBM Fusion services.

Deploying IBM Fusion using CLI

Use the instructions to deploy the IBM Fusion using the command-line interface (CLI).

Procedure

1. Log in to the OpenShift® terminal using the following command.

```
oc login --token=${TOKEN_VALUE} --server=${SERVER_NAME}
```

2. Create the YAML to deploy the IBM Fusion.

```
cat << EOF > deployfusion.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-spectrum-fusion-ns
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: isf-og
  namespace: ibm-spectrum-fusion-ns
spec:
  targetNamespaces:
  -  ibm-spectrum-fusion-ns
  upgradeStrategy: Default
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image:  icr.io/cpopen/ibm-operator-catalog:latest
  displayName: IBM Operator Catalog
  publisher: IBM
  updateStrategy:
   registryPoll:
    interval: 45m
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: isf-operator
  namespace: ibm-spectrum-fusion-ns
spec:
  channel: v2.0
  name: isf-operator
  sourceNamespace: openshift-marketplace
  source: ibm-operator-catalog
  installPlanApproval: Manual
EOF
```

3. Run the following command to apply the YAML created in the previous step.

```
oc apply -f deployfusion.yaml
```

Sample output:

```
namespace/ibm-spectrum-fusion-ns created
operatorgroup.operators.coreos.com/isf-og created
catalogsource.operators.coreos.com/ibm-operator-catalog created
subscription.operators.coreos.com/isf-operator created
```

4. Run the following command to verify whether the IBM Fusion is deployed.

```
oc get pods -n ibm-spectrum-fusion-ns
```

Sample output:

```
NAME                                                        READY   STATUS    RESTARTS   AGE
fusion-console-8fcc98dd7-slp8g                              1/1     Running   0          3m36s
isf-application-operator-controller-manager-5796c55576-9p76j   2/2     Running   0          3m50s
isf-cns-operator-controller-manager-6f6c5ff5b7-kfx87        2/2     Running   0          3m54s
isf-data-protection-operator-controller-manager-7fbd4489f-vqn9j   2/2     Running   0          3m58s
isf-prereq-operator-controller-manager-76b8547ff5-hj9zw     2/2     Running   0          4m2s
isf-proxy-6bb976f5cf-9n7fn                                  1/1     Running   0          3m36s
isf-proxy-6bb976f5cf-xt25d                                  1/1     Running   0          3m36s
isf-serviceability-operator-controller-manager-7dc8bbc7fb-zkdt8   2/2     Running   0          3m37s
isf-ui-dep-85688bb65f-g8zhg                                 1/1     Running   0          3m36s
isf-ui-dep-85688bb65f-hhd8k                                 1/1     Running   0          3m36s
isf-ui-operator-controller-manager-6b78f885c4-2jcmh         2/2     Running   0          3m41s
isf-update-operator-controller-manager-76b89b777f-lbdvp     2/2     Running   0          3m45s
```

5. After the IBM Fusion installation, create the YAML to accept the license.

```
cat << EOF > accept_license.yaml
apiVersion: prereq.isf.ibm.com/v1
kind: SpectrumFusion
```

```
metadata:
  name: spectrumfusion
  namespace: ibm-spectrum-fusion-ns
spec:
  license:
    accept: true
EOF
```

6. Run the following command to apply the YAML created in the previous step to accept the license.

   ```
   oc apply -f accept_license.yaml
   ```

   Sample output:

   ```
   spectrumfusion.prereq.isf.ibm.com/spectrumfusion created
   ```

7. Run the following command to check whether the CR is created successfully.

   ```
   oc get SpectrumFusion -n ibm-spectrum-fusion-ns
   ```

   Sample output:

   ```
   NAME              AGE
   spectrumfusion    4m53s
   ```

8. Run the following command to check the IBM Fusion service definition.

   ```
   oc get FusionServiceDefinitions -n ibm-spectrum-fusion-ns
   ```

   Sample output:

   ```
   NAME                                  AGE
   data-cataloging-service-definition    18m
   data-foundation-service               18m
   ibm-backup-restore-agent-service      18m
   ibm-backup-restore-service            18m
   ```

Validating IBM Fusion installation

Steps to confirm the success of the installation.

- Health status of individual services
- Verify the IBM Fusion installation
- Verify the Global data platform installation
- Verify the Backup & Restore installation
- Verify the Fusion Data Foundation
- Verify the Data Cataloging installation
- Verify the IBM Fusion CR

Health status of individual services

In the Services page, check the health status of your installed services. The services must be in running or healthy state.

Verify the IBM Fusion installation

Verify pods from the OpenShift® Container Platform console:

1. Click View Operator to view the installed operators in the selected namespace. In this example, it is **ibm-spectrum-fusion-ns** namespace.
2. Go to Workloads > Pods.
3. Select the namespace where you installed IBM Fusion.
4. Check whether the following pods are running successfully:
   - **isf-application-operator-controller-manager**
   - **isf-cns-operator-controller-manager**
   - **isf-data-protection-operator-controller-manager**
   - **isf-prereq-operator-controller-manager**
   - **isf-serviceability-operator-controller-manager**
   - **isf-ui-operator-controller-manager**
   - **spp-dp-controller-manager**
   - **callhomeclient**
   - **isf-proxy**
   - **isf-ui-dep**
   - **logcollector**
   - **eventmanager**
   - **trapserver**
   - **isf-update-operator-controller-manager**

Alternatively, you can verify the installation by running the oc commands:

```
oc get pods -n ibm-spectrum-fusion-ns
```

A sample result of the oc command is as follows:

```
[root@vm-1527 ~]# oc get pods
NAME                                                              READY   STATUS    RESTARTS   AGE
```

```
callhomeclient-75cb5d95b9-f6nmk                                  1/1     Running   0          130m
isf-application-operator-controller-manager-7b95c84d6d-fqpbp     2/2     Running   0          137m
isf-cns-operator-controller-manager-6d8754cfd4-lnnlz             2/2     Running   1          137m
isf-data-protection-operator-controller-manager-986bfc476-pzsv6  2/2     Running   1          137m
isf-prereq-operator-controller-manager-597c68796-62cnk           2/2     Running   1          137m
isf-proxy-5bbcbd786b-hhzsj                                       1/1     Running   0          134m
isf-serviceability-operator-controller-manager-54c555676c-7l4pq  2/2     Running   0          137m
isf-ui-dep-8486747fff-xjhkw                                      1/1     Running   0          134m
isf-ui-operator-controller-manager-6b6678dc9c-sqxhk              2/2     Running   1          137m
logcollector-76f696c4bb-lppmg                                    1/1     Running   0          130m
spp-dp-controller-manager-84b9d984d9-wjwnn                       1/1     Running   1          137m
isf-update-operator-controller-manager-97f666b4f-tc47k           1/1     Running   0
```

Make sure all the pods are up and running.

Verify the IBM Fusion instances as follows:

1. Click Operators > Install operators. Select the namespace in which IBM Fusion was installed from the Project tab.
2. Select the IBM Fusion.
3. Go to All instance tab. It lists the IBM Fusion instances that are created. Ensure that the status shows completed for all instances.
   The following instances get created and can be validation points for IBM Fusion post installation.

   - `spectrumfusion` CR instance of Kind `SpectrumFusion`
   - `data-cataloging-service-definition` CR instance of Kind `FusionServiceDefinition`
   - `data-foundation-service` CR instance of Kind `FusionServiceDefinition`
   - `ibm-backup-restore-agent-service` CR instance of Kind `FusionServiceDefinition`
   - `ibm-backup-restore-service` CR instance of Kind `FusionServiceDefinition`
   - Cluster object of kind `Cluster`, the name of the object can be the DNS entry of the cluster.
     For example: `apps.sds-auto3.cp.fyre.ibm.com`
   - CR instance objects of Kind `Application` with names:

     `ibm-backup-restore`
     `ibm-data-cataloging`
     `ibm-spectrum-fusion-ns`
     `rook-ceph`
   - CR instance objects of Kind `BackupStorageLocation` with names:

     `in-place-snapshot`
     `isf-dp-inplace-snapshot`
   - Recipe object with name `fusion-cr-backup`

To open the IBM Fusion user interface in the OpenShift Container Platform web console, click the Applications icon in the title bar and select IBM Fusion.
Note: You can use IBM Fusion user interface only when the installation of IBM Fusion is completed successfully.

Verify the Global data platform installation

Applicable for Red Hat® OpenShift Container Platform 4.21 or later version:
Note: Ensure that the Fusion Data Foundation service is configured with IBM Storage Scale before validation.
In the Red Hat OpenShift Container Platform web console, go to Workloads > Pods and verify the following:

1. The `ibm-spectrum-scale-controller-manager` is present in the `ibm-spectrum-scale` project.
2. All pods in the `ibm-spectrum-scale` project have a status of `Running`.

Applicable for Red Hat OpenShift Container Platform 4.20 or earlier version:
Note: Ensure that the Global Data Platform service is installed and configured before validation.

Verify whether the IBM Storage Scale installed successfully.

Note: Before you validate, ensure that the IBM Storage Scale is used as the storage provider.
Note: The `ibm-spectrum-scale` namespace does not have any pods until the remote mount is configured from IBM Fusion UI, that is, the remote mount filesystem is created.

1. Run the following command to validate the IBM Storage Scale installation.

   ```
   oc describe scalemanager -n ibm-spectrum-fusion-ns
   ```
2. Check whether the status at the end of CR reflects as completed.
3. If the IBM Storage Scale installation is still in progress, the status continues to show Installing. Wait and ensure that the IBM Storage Scale installation completes.

Verify the IBM Storage Scale pods as follows:

1. Verify the IBM Storage Scale pods from the OpenShift Container Platform console:
   a. Go to Workloads > Pods.
   b. Select `ibm-spectrum-scale`, `ibm-spectrum-scale-csi`, and `ibm-spectrum-scale-operator` namespaces. It lists all the pods and makes sure that all pods are running.
   Note: The `ibm-spectrum-scale` namespace does not have any pods until the remote mount is configured from IBM Fusion UI.
2. Verify the IBM Storage Scale deployments and pods from the IBM Fusion CLI.
   After you successfully remote mount the filesystem, run the following commands to validate it.

   a. To list all scale pods, run the following command:

      ```
      oc -n ibm-spectrum-scale get pods
      ```

      Example output:

```
./oc get pods -n ibm-spectrum-scale
NAME                                READY   STATUS                  RESTARTS   AGE
ibm-spectrum-scale-gui-0            4/4     Running                 0          8m30s
ibm-spectrum-scale-gui-1            4/4     Running                 0          4m30s
ibm-spectrum-scale-pmcollector-0    2/2     Running                 0          8m
ibm-spectrum-scale-pmcollector-1    2/2     Running                 0          6m29s
worker0                             2/2     Running                 0          8m30s
worker1                             2/2     Running                 0          8m30s
worker2                             2/2     Running                 0          8m30s
```

The output lists two GUI pods and a daemon pod for each worker node.

b. Validate pods in **ibm-spectrum-scale-dns**:

```
oc get pods -n ibm-spectrum-scale-dns
```

Example output:

```
NAME                                READY   STATUS                  RESTARTS     AGE
coredns-98mqn                        1/1     Running                 0          8m43s
coredns-9ngqc                        1/1     Running                 0          8m43s
coredns-h658w                        1/1     Running                 0          8m43s
coredns-nw8dj                        1/1     Running                 0          8m43s
2/2     Running                 0          8m30s
coredns-pqh5f                        1/1     Running                 0          8m43s
coredns-qzgdk                        1/1     Running                 0          8m43s
```

c. To list all the CSI pods, run the following command:

```
oc get pods -n ibm-spectrum-scale-csi
```

Example output:

```
NAME                                                  READY   STATUS              RESTARTS   AGE
ibm-spectrum-scale-csi-attacher-5d9684696b-9fvdq      1/1     Running             0          3m33s
ibm-spectrum-scale-csi-attacher-5d9684696b-mp8qn      1/1     Running             0          3m33s
ibm-spectrum-scale-csi-f6n17                          3/3     Running             0          3m33s
ibm-spectrum-scale-csi-j22nt                          3/3     Running             0          3m33s
ibm-spectrum-scale-csi-operator-8488bcd9c-vj2bz       1/1     Running             0          102m
ibm-spectrum-scale-csi-provisioner-5d5fbdffbb-qbgx9   1/1     Running             0          3m33s
ibm-spectrum-scale-csi-resizer-7d78944d7b-m74nm       1/1     Running             0          3m33s
ibm-spectrum-scale-csi-snapshotter-7f6dc898f-m5zdn    1/1     Running             0          3m33s
ibm-spectrum-scale-csi-sw9gk                          3/3     Running             0          3m33s
```

d. To show the IBM Fusion defined scale cluster, run the following command:

```
oc -n ibm-spectrum-fusion-ns get scalecluster -o yaml
```

e. To check the scale filesystem CR results, run the following command:

```
oc -n ibm-spectrum-scale get filesystem -o yaml
```

f. To check the scale remote cluster CR result, run the following command:

```
oc -n ibm-spectrum-scale get remotecluster -o yaml
```

g. Run the oc command and check whether Daemon.status.clusterID has a value or not.

```
oc -n ibm-spectrum-scale get daemon -oyaml
```

Verify the Backup & Restore installation

Note: Ensure that the Backup & Restore IBM Fusion service is enabled before validation.

Verify that the Backup & Restore IBM Fusion service operators from the OpenShift Container Platform web console:

1. Go to Operators > Installed operators from OpenShift Container Platform web console.
2. Select the project as **ibm-backup-restore**.
3. Verify that the following operators show the status as **succeeded**.
   - Red Hat Integration - AMQ Streams
   - IBM Fusion Backup and Restore Server
   - IBM Fusion Backup and Restore Agent
   - OADP Operator
4. Alternatively, you can verify the status of operators by running the following oc command:
   Note: More operators appear in the command line output than in the web console.

   ```
   oc get csv -n ibm-backup-restore
   ```

   A sample result of the oc command is as follows:

   ```
   # oc get csv -n ibm-backup-restore
   NAME                              DISPLAY                                                VERSION   REPLACES
   PHASE
   amqstreams.v2.6.0-2               AMQ Streams                                            2.6.0-2
   amqstreams.v2.6.0-1    Succeeded
   guardian-dm-operator.v2.8.0       IBM Fusion Backup and Restore Data Mover        2.8.0
   Succeeded
   guardian-dp-operator.v2.8.0       IBM Fusion Backup and Restore Data Protection   2.8.0
   Succeeded
   guardian-mongo-operator.v2.8.0    IBM Fusion Backup and Restore Mongo             2.8.0
   Succeeded
   ```

```
ibm-dataprotectionagent.v2.8.0    IBM Fusion Backup and Restore Agent             2.8.0
Succeeded
ibm-dataprotectionserver.v2.8.0   IBM Fusion Backup and Restore Server            2.8.0
Succeeded
oadp-operator.v1.3.1              OADP Operator                                           1.3.1     oadp-
operator.v1.3.0   Succeeded
redis-operator.v2.8.0             IBM Fusion Backup and Restore Redis             2.8.0
Succeeded
```

Ensure that the status shows **Succeeded**.

Verify the Backup & Restore pods from the OpenShift Container Platform console:

1. Go to Workloads > Pods.
2. Select the namespace, where you installed IBM Fusion Backup & Restore. In this case, select **ibm-backup-restore** namespace.
   It lists all the pods. Ensure that all pods are running.
3. Verify whether the following pods are running successfully:

```
amq-streams-cluster-operator
applicationsvc
backup-location-deployment
backup-service
backuppolicy-deployment
guardian-dm-controller-manager
guardian-dp-operator-controller-manager
guardian-kafka-cluster-entity-operator
guardian-kafka-cluster-kafka-0
guardian-kafka-cluster-kafka-1
guardian-kafka-cluster-kafka-2
guardian-kafka-cluster-zookeeper-0
guardian-kafka-cluster-zookeeper-1
guardian-kafka-cluster-zookeeper-2
guardian-minio-0
guardian-mongo-operator-controller-manager
ibm-dataprotectionagent-controller-manager
ibm-dataprotectionserver-catalog-ibm-backup-restore
ibm-dataprotectionserver-controller-manager
ibm-backup-restoreagent-controller-manager
ibm-backup-restoreserver-controller-manager
job-manager
mongodb-0
mongodb-1
mongodb-2
mongodb-ab-0
openshift-adp-controller-manager
redis-master-0
redis-operator-controller-manager
redis-replicas-0
redis-replicas-1
redis-replicas-2
transaction-manager
velero
```

4. Alternatively, you can verify the installation by running the following oc command:

```
oc get pods -n ibm-backup-restore
```

A sample result of the oc command is as follows:

```
NAME                                                              READY   STATUS      R
amq-streams-cluster-operator-v2.3.0-1-7d6fb79d84-jdkfh            1/1     Running     0
applicationsvc-55c9b4d6c9-6hdv7                                   1/1     Running     0
b0f64f9161e0882f278dde2eaa1ea9677f4a230a29180fcf21fc665761hvvxz   0/1     Completed   0
backup-location-deployment-6b565b856c-j4vjc                       1/1     Running     0
backup-service-54bf9988f6-47bpv                                   1/1     Running     0
backuppolicy-deployment-b997cc9bf-dfwnh                           1/1     Running     0
bc6176f08ef686ccde24395724b77ea07a586a0bd1fa27ebfd5d704d0dxv9pl   0/1     Completed   0
e349f7c16f02ad6c0c31e41ba2fb1750d5154b58537224d239fe47508872cj5   0/1     Completed   0
f3dd0cbe8cb98614cf163fc5372733148236ea2bdeb5efc6a5d5afe4c085qlk   0/1     Completed   0
ff53c6d827e0fd610a4392c4f9411beb0c785572d2fca1bda57e208650bwv2m   0/1     Completed   0
ffacc5cdaa1e0aa4f3b3c0021f3c87931aa7422f8415303d95457febc8nmzk7   0/1     Completed   0
guardian-dm-controller-manager-64d57bf9ff-28dqj                   2/2     Running     0
guardian-dp-operator-controller-manager-6f6d55f6f7-fhndb          2/2     Running     0
guardian-kafka-cluster-entity-operator-b59d699f7-5qxt8            3/3     Running     0
guardian-kafka-cluster-kafka-0                                    1/1     Running     0
guardian-kafka-cluster-kafka-1                                    1/1     Running     0
guardian-kafka-cluster-kafka-2                                    1/1     Running     0
guardian-kafka-cluster-zookeeper-0                                1/1     Running     0
guardian-kafka-cluster-zookeeper-1                                1/1     Running     0
guardian-kafka-cluster-zookeeper-2                                1/1     Running     0
guardian-minio-0                                                  1/1     Running     0
guardian-mongo-operator-controller-manager-6f47776cb4-s6tkm       2/2     Running     0
ibm-dataprotectionagent-controller-manager-54d66f7975-lgdgh       2/2     Running     0
ibm-dataprotectionserver-catalog-ibm-backup-restore-k967t         1/1     Running     0
ibm-dataprotectionserver-controller-manager-749554d89f-q6gmx      2/2     Running     0
job-manager-859484bfc5-fzpt8                                      1/1     Running     0
mongodb-0                                                         2/2     Running     1
mongodb-1                                                         2/2     Running     1
mongodb-2                                                         2/2     Running     1
mongodb-ab-0                                                      1/1     Running     0
openshift-adp-controller-manager-59fb9f86f4-gdbhb                 1/1     Running     0
redis-master-0                                                    1/1     Running     0
redis-operator-controller-manager-647cf89ff7-wl8w2                2/2     Running     0
redis-replicas-0                                                  1/1     Running     0
```

```
redis-replicas-1                                                  1/1     Running     0
redis-replicas-2                                                  1/1     Running     0
transaction-manager-5d9b59cf9f-hdzjm                              2/2     Running     0
velero-777d65c9b7-455fv                                           1/1     Running     0
```

Verify the Backup & Restore IBM Fusion service installation status from the OpenShift Container Platform web console:

1. Go to Operators > Installed operators from OpenShift Container Platform web console.
2. Select the namespace as **ibm-backup-restore**.
3. Select IBM Fusion Backup & Restore server.
4. Click the Data Protection server tab and select Data Protection server.
5. Select the YAML tab.
6. In the status section, makes sure the following:
   - **HealthStatuses** Shows the health status of all components listed. It may take 5 minutes or more for all components to show up as healthy.
   - Ensure the install status shows Complete and **progressPercentage** as 100.
7. Alternatively, you can verify the installation status by running the following oc command:

   ```
   oc describe dataprotectionserver dataprotectionserver -n ibm-backup-restore
   ```

   A sample result of the **oc** command is as follows:

   ```
   Status:
     Health Statuses:
       Service Name:  applicationservice
       Status:        Healthy
       Service Name:  backuplocation
       Status:        Healthy
       Service Name:  backuppolicy
       Status:        Healthy
       Service Name:  backupservice
       Status:        Healthy
       Service Name:  jobmanager
       Status:        Healthy
       Service Name:  backupagent
       Status:        Healthy
       Service Name:  mongo
       Status:        Healthy
       Service Name:  redis
       Status:        Healthy
       Service Name:  kafka
       Status:        Healthy
     Install Status:
       Progress Percentage:  100
       Retry On Failure:     false
       Status:               Completed
     Installed Version:      2.5.1
     Upgrade In Progress:    false
     Upgrade Status:
       Retry On Failure:  false
   ```

8. Check the status section at the end of the CR and make sure the following:
   - The Install Status section shows the status of the install along with the percentage complete.
   - The install status shows the install status as Completed when the installation is successfully completed.
   - The health status section lists and shows components health status as healthy.
   - Note: Individual component health statuses may show **Unknown** or **Degraded** for up to five minutes, and show a healthy status when installation is complete.

Verify the Fusion Data Foundation

Verify the Fusion Data Foundation installation was completed successfully from the OpenShift Container Platform console:

1. Go to Installed Operators to check whether the Fusion Data Foundation operator is listed and status is Succeeded.
2. Verify pods from the Red HatOpenShift Container Platform web console:
   a. Go to Workloads > Pods.
   b. Select the **openshift-storage** namespace, where the Fusion Data Foundation is installed.
   c. Check whether the following pods are running successfully:
      - **odf-console**
      - **odf-operator-controller-manager**
3. Alternatively, you can verify the installation by running the OC commands:

   ```
   oc get pod -n openshift-storage
   ```

   A sample result of the OC command is as follows:

   ```
   NAME                                               READY   STATUS              RESTARTS   AGE
   odf-console-849f64fdd7-rqd97                       1/1     Running             0          3m3s
   odf-operator-controller-manager-5bd4c85d6b-q8g2f   1/1     Running             0          3m3s
   ```

4. Ensure that all the pods are up and running.
5. Verify the Fusion Data Foundation status as follows:
   a. Click Operators > Install operators. Select the namespace in which IBM Fusion was installed from the Project tab.
   b. Select the IBM Fusion.
   c. Go to Fusion Service Instance tab and click odfmanager from the list.
   d. Open **YAML** tab and ensure that **.status.health** is Healthy.
6. Alternatively, you can verify the status by running the OC commands:

   ```
   oc get fusionserviceinstances.service.isf.ibm.com -n ibm-spectrum-fusion-ns odfmanager -o jsonpath='{.status.health}
   {"\n"}'
   ```

7. Verify the local storage operator pods when backing storage type is local as follows:
   a. Click Operators > Install operators. Select the namespace in which IBM Fusion was installed from the Project tab.
   b. Select the IBM Fusion.
   c. Go to Fusion Service Instance tab and click odfmanager from the list.
   d. Click YAML tab and check the **backingStorageType**.

```
- name: backingStorageType
  provided: true
  value: Local
```

   e. Alternatively, you can verify the backing storage type by running the oc commands.

```
oc get fusionserviceinstances.service.isf.ibm.com -n ibm-spectrum-fusion-ns odfmanager -o
jsonpath='{.spec.parameters}{"\n"}'
```

   f. If the backing storage type is local, then verify the local storage operator.
   Go to Installed Operators to check whether the Local Storage operator is listed and its status is Succeeded.

   Note: The green check mark indicates that the local storage operator is installed.
   g. Verify pods from the Red HatOpenShift Container Platform web console:
      i. Go to Workloads > Pods.
      ii. Select the **openshift-local-storage** namespace, where the local storage operator is installed.
      iii. Check whether the following pods are running successfully: **local-storage-operator-xxxxx**
      For example: **local-storage-operator-75c55cf57d-pfjcp**

   h. Ensure that all pods are running.
   i. Alternatively, you can verify the installation by running the oc commands:

```
oc get pod -n openshift-local-storage
```

   A sample result of the oc command is as follows:

```
NAME                                  READY   STATUS    RESTARTS   AGE
diskmaker-discovery-4pw9r             2/2     Running   0          30m
diskmaker-discovery-b4s58             2/2     Running   0          30m
diskmaker-discovery-cc77p             2/2     Running   0          30m
diskmaker-discovery-cqx6h             2/2     Running   0          30m
diskmaker-discovery-kzxrs             2/2     Running   0          30m
diskmaker-discovery-vnhdl             2/2     Running   0          30m
diskmaker-discovery-vvkpd             2/2     Running   0          30m
local-storage-operator-685586b5d6-67nbc 1/1   Running   0          30m
```

Verify the Data Cataloging installation

Verify the IBM Data Cataloging installation was completed successfully from the OpenShift Container Platform web console:

1. Run the following command to verify the installed operators.

```
oc -n ibm-data-cataloging get csv
```

```
NAME                                                DISPLAY                VERSION              REPLACES                PH
amqstreams.v2.6.0-2                                 AMQ Streams            2.6.0-2              amqstreams.v2.6.0-1     Su
db2u-operator.v110509.0.1                           IBM Db2                110509.0.1                                   Su
ibm-spectrum-discover-operator.v216.0.0-1715012933  IBM Storage Discover   216.0.0-1715012933                           Su
```

2. Run the following command to verify IBM Data Cataloging status.

```
oc -n ibm-data-cataloging get isd
```

```
Expected output
NAME                            READY   SUMMARY                        ERROR   AGE   HEALTHSTATUS
data-cataloging-service-instance   True    Awaiting next reconciliation           9d    Healthy
```

3. Run the following command to verify that all Persistent Volume Claims are bounded.

```
oc -n ibm-data-cataloging get pvc
```

```
Expected output
NAME                        STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
activelogs-c-isd-db2u-0     Bound    pvc-7e98eead-9a38-48b6-a880-16d730120796   150Gi      RWO            ocs-storageclus
activelogs-c-isd-db2u-1     Bound    pvc-7af71786-1529-4b3c-bbd1-b5d1ee733d33   150Gi      RWO            ocs-storageclus
c-isd-meta                  Bound    pvc-1ba20699-ddb8-4600-bf9f-4d637b14da49   50Gi       RWX            ocs-storageclus
data-c-isd-db2u-0           Bound    pvc-488540db-c951-4f48-b65c-814bf634ca33   300Gi      RWO            ocs-storageclus
data-c-isd-db2u-1           Bound    pvc-56a3105a-347e-4b55-b11a-d0a1d351d622   300Gi      RWO            ocs-storageclus
data-isd-sasl-kafka-0       Bound    pvc-3924c9cc-173d-4244-90fa-a234325dc04b   100Gi      RWO            ocs-storageclus
data-isd-sasl-zookeeper-0   Bound    pvc-3c66397b-ac03-4ed8-841a-e33d83355a23   64Gi       RWO            ocs-storageclus
data-isd-ssl-kafka-0        Bound    pvc-ce0ebeaf-d2b6-4e40-8c31-e31d45a208e5   100Gi      RWO            ocs-storageclus
data-isd-ssl-zookeeper-0    Bound    pvc-cd3fdd20-343b-49ca-ad1a-690f8eec3e60   64Gi       RWO            ocs-storageclus
isd-backup                  Bound    pvc-2e077c8e-e992-4e1a-8b98-d24bfdbb5343   250Gi      RWX            ocs-storageclus
isd-data                    Bound    pvc-b98e124f-1f98-43e5-9bf6-c8758c681192   250Gi      RWX            ocs-storageclus
tempts-c-isd-db2u-0         Bound    pvc-7fdc9ac5-9158-48ac-bad3-3c24590f1f2b   50Gi       RWO            ocs-storageclus
tempts-c-isd-db2u-1         Bound    pvc-a2adb668-a362-4f61-8ed3-ae103f9b2cd3   50Gi       RWO            ocs-storageclus
```

4. Run the following command to verify Db2u cluster detailed progress.

```
oc -n ibm-data-cataloging get formations.db2u.databases.ibm.com isd -o go-template='{{range .status.components}}
{{printf "%s,%s,%s\n" .kind .name .status.state}}{{end}}' | column -s, -t
```

```
Expected output
account                account-ibm-data-cataloging-isd  OK
PersistentVolumeClaim  c-isd-meta                       OK
```

```
secret                  c-isd-sshkeys-db2uhausr          OK
secret                  c-isd-sshkeys-db2uadm            OK
secret                  c-isd-sshkeys-db2instusr         OK
secret                  c-isd-ldappassword               OK
secret                  c-isd-ldapblueadminpassword      OK
secret                  c-isd-instancepassword           OK
secret                  c-isd-db2u-lic                   OK
secret                  c-isd-certs-wv-rest              OK
secret                  c-isd-certs-db2u-api             OK
configmap               c-isd-db2uconfig                 OK
configmap               c-isd-db2regconfig               OK
configmap               c-isd-db2dbmconfig               OK
configmap               c-isd-db2dbconfig                OK
service                 c-isd-ldap                       OK
Deployment              c-isd-ldap                       OK
service                 c-isd-tools                      OK
Deployment              c-isd-tools                      OK
configmap               c-isd-db2u-api                   OK
service                 c-isd-db2u-engn-svc              OK
service                 c-isd-db2u-head-engn-svc         OK
Job                     c-isd-instdb                     OK
service                 c-isd-etcd                       OK
StatefulSet             c-isd-etcd                       OK
service                 c-isd-db2u-rest-svc              OK
networkpolicy           c-isd-rest-ext                   OK
Deployment              c-isd-rest                       OK
service                 c-isd-db2u-graph-svc             OK
networkpolicy           c-isd-graph-ext                  OK
Deployment              c-isd-graph                      OK
service                 c-isd-db2u-internal              OK
service                 c-isd-db2u                       OK
StatefulSet             c-isd-db2u                       OK
networkpolicy           c-isd                            OK
networkpolicy           c-isd-ext                        OK
Job                     c-isd-restore-morph              OK
```

5. Run the following command to verify the Db2u cluster CR status.

```
oc -n ibm-data-cataloging get db2ucluster
```

```
Expected output
NAME   STATE   MAINTENANCESTATE   AGE
isd    Ready   None               9d
```

6. Run the following command to verify that the ports 50000 and 50001 are listing.

```
oc exec -n ibm-data-cataloging -it $(oc -n ibm-data-cataloging get po --no-headers --show-labels=true --selector
name=dashmpp-head-0 | awk '{print $1}') -- su - db2inst1 -c 'netstat -ntlp'
```

```
Expected output
Defaulted container "db2u" out of: db2u, init-labels (init), init-kernel (init)
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 10.254.4.11:60000       0.0.0.0:*               LISTEN      23707/db2sysc 0
tcp        0      0 10.254.4.11:60001       0.0.0.0:*               LISTEN      23712/db2sysc 1
tcp        0      0 0.0.0.0:9443            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:50022           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:50000           0.0.0.0:*               LISTEN      23707/db2sysc 0
tcp        0      0 0.0.0.0:50001           0.0.0.0:*               LISTEN      23707/db2sysc 0
tcp6       0      0 :::50052                :::*                    LISTEN      -
tcp6       0      0 :::50022                :::*                    LISTEN      -
```

7. Run the following command to verify that workload is PUREDATA_OLAP.

```
oc exec -n ibm-data-cataloging -c db2u -it $(oc -n ibm-data-cataloging get po --no-headers --show-labels=true --
selector name=dashmpp-head-0 | awk '{print $1}') -- sudo su - db2inst1 -c "db2set -all | grep DB2_WORKLOAD= | awk
'{print \$2}'"
```

```
Expected output
DB2_WORKLOAD=PUREDATA_OLAP
```

8. Run the following command to verify the database encryption configurations.

```
oc -n ibm-data-cataloging exec -c db2u $(oc -n ibm-data-cataloging get po --no-headers --show-labels=true --selector
name=dashmpp-head-0 | awk '{print $1}') -- su - db2inst1 -c "db2 get db cfg for bludb | grep 'Encrypted database'"
```

```
 Encrypted database                                  = YES
```

9. Run the following command to verify the Kafka CRs readiness.

```
oc -n ibm-data-cataloging get kafka
```

```
Expected output
NAME       DESIRED KAFKA REPLICAS   DESIRED ZK REPLICAS   READY   WARNINGS
 isd-sasl   1                        1                     True
 isd-ssl    1                        1                     True
```

10. Run the following command to verify that the IBM® Spectrum Discover pods are running.

```
oc -n ibm-data-cataloging get pod -l component=discover
```

```
Expected output
 NAME                                    READY   STATUS      RESTARTS     AGE
isd-api-7564475d98-cs6nx                 1/1     Running     0            9d
```

```
isd-auth-5fcb8bd95f-jvzwp                    1/1     Running     0             9d
isd-backup-restore-689fc64f7-pgx22           1/1     Running     0             9d
isd-connmgr-7d4cdb4cc7-xnkzm                 1/1     Running     1 (9d ago)    9d
isd-consumer-ceph-le-5f9b5d4676-4fsgg        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-9k9cs        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-bphqp        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-cc2pz        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-chlvk        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-fchp5        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-jlbrw        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-l78cc        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-lpcsn        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-n2wqt        1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-54qm9         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-6hsq9         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-9p4p5         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-dhlgc         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-hr76h         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-mkvx8         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-tvscz         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-vrrd4         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-w59c2         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-xqncl         1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-4klcc       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-8r6w7       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-fqs5m       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-fwb8p       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-h78g5       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-lvwl5       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-nhznb       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-q218b       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-z6tfj       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-zv2zw       1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-5mrrg      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-8rphj      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-b5dbv      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-b5lv2      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-d9gbm      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-f5rxf      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-m8nw6      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-ntfbb      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-ql55x      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-qvggm      1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-6s75x    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-845bx    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-8kmnf    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-gntbt    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-mlnfb    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-p9lpt    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-rckm8    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-vkmpd    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-vmh8x    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-zm4gf    1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-6dw9h       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-9nhsz       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-dnwpw       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-dz2mf       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-h2ffj       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-hdrxg       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-nf779       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-pf6n4       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-q8gqf       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-xzmvh       1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-49vkb     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-54pb8     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-5sr89     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-98lm9     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-d4zlj     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-fc82j     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-mnsn2     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-r2kc6     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-tglsz     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-zbhgv     1/1     Running     0             9d
isd-contentsearchagent-5d7bfbd887-r487x      1/1     Running     1 (9d ago)    9d
isd-db-schema-xm5cn                          0/1     Completed   0             9d
isd-db2whrest-6c6bbfcbc6-9v6v2               1/1     Running     1 (9d ago)    9d
isd-generate-license-6g94v                   0/1     Completed   0             9d
isd-importtags-64498dbdd8-5kqpb              1/1     Running     1 (9d ago)    9d
isd-keystone-7dc84b69f5-sw6nn                1/1     Running     0             9d
isd-license-check-27982200-c9gkm             0/1     Completed   0             2d21h
isd-license-check-27983640-ldxv2             0/1     Completed   0             45h
isd-license-check-27985080-rhzjl             0/1     Completed   0             21h
isd-policyengine-598f565867-zg5sn            1/1     Running     1 (9d ago)    9d
isd-producer-ceph-le-76b9885854-557b7        1/1     Running     0             9d
isd-producer-cos-le-85d446c9d8-zvzmg         1/1     Running     0             9d
isd-producer-cos-scan-d584f646d-2t669        1/1     Running     0             9d
isd-producer-file-scan-5586b48fd-zkt7q       1/1     Running     0             9d
isd-producer-protect-scan-7b66bb7c64-wz6qq   1/1     Running     0             9d
isd-producer-scale-le-8d7bc64b4-nvqbl        1/1     Running     0             9d
isd-producer-scale-scan-5df647956-qfl6p      1/1     Running     0             9d
isd-proxy-5f4d7dd8ff-44fwt                   1/1     Running     0             9d
isd-reports-c6kmq                            0/1     Completed   0             9d
isd-scaleafmdatamover-85cc44b566-644vx       1/1     Running     1 (9d ago)    9d
isd-scaleilmdatamover-6b86dd7b58-j28sn       1/1     Running     1 (9d ago)    9d
isd-sdmonitor-6fffc999b7-s2srd               1/1     Running     0             9d
isd-tikaserver-79548bdb4-ssft5               1/1     Running     0             9d
```

```
isd-ui-backend-f6c4b65d7-dwzrs              1/1     Running     0            9d
isd-ui-frontend-9659fc847-vcbmv             1/1     Running     0            9d
```

11. Run the following command to verify the console route availability.

```
oc -n ibm-data-cataloging get route console
```

```
Expected output
NAME      HOST/PORT                                          PATH   SERVICES   PORT    TERMINATION   WILDCARD
console   console-ibm-data-cataloging.apps.ocp2.vmlocal             isd-svc    <all>   reencrypt     None
```

Verify the IBM Fusion CR

IBM Fusion installation status can also be checked through IBM Fusion CR.

1. Important: The IBM Fusion namespace **ibm-spectrum-fusion-ns** is used in the following steps for your reference. If IBM Fusion is installed in a different namespace, replace the namespace with your namespace.
   Run the following command and export IBM Fusion namespace as environmental variable.

   ```
   export FUSION_NS="ibm-spectrum-fusion-ns"
   ```

2. Run the command to validate the IBM Fusion installation status.

   ```
   oc describe SpectrumFusion/spectrumfusion -n $FUSION_NS
   ```

   Where the **SpectrumFusion** is the kind of the CR and **spectrumfusion** is name of the CR.

3. Go to OpenShift Container Platform console, go to Operators > Installed operators.
4. Select the IBM Fusion namespace from the Project drop down and select the Fusion tab.
   It shows the CR kind and name of the CR.

5. This CR show status and version of all IBM Fusion services and the IBM Fusion Install status.

```
Spec:
   Global Data Platform:
      Enable: true
   License:
      Accept: true
```

```
Status:
    Global Data Platform Status:
       Service Enabled:               true
       Health:                        Healthy
       Install Status:                Completed
       Install Status Message:        IBM Spectrum Scale storage installation is succeeded.
       Install Status Message Code:   BMYSS0003
       Is CRC Rendered:               false
       Is Deployable:                 true
       Is Machine config Rendered:    false
       Is supported:                  true
       Is Upgrade Failed:             false
       Is Upgrade In Progress:        false
       Max Available Version:         5.2.0
       Progress Percentage:           100
       Upgrade Available:             false
       Version:                       5.2.0
  Events: <none>
```

Uninstalling IBM Fusion

Steps to uninstall IBM Fusion from Red Hat® OpenShift® web console.

Before you begin

- Uninstall all IBM Fusion services before you uninstall IBM Fusion.
- Ensure that you delete all the connection CRDs. For the procedure to delete, see Disabling the connection.

Procedure

1. Export IBM Fusion namespace as an environment variable.
   Note: If the IBM Fusion is installed in a different namespace, replace the existing namespace:

   ```
   export FUSION_NS="ibm-spectrum-fusion-ns"
   ```

2. Run the following script:

```
oc project "$FUSION_NS"
echo "Fusion namespace: $FUSION_NS"
echo "Labeling namespace $FUSION_NS with uninstalling=true"
oc label namespace "$FUSION_NS" uninstalling=true --overwrite

oc delete isfproxydeps isf-proxy
oc delete isflcdeps logcollector
oc delete isfchdeps callhomeclient
oc delete oauthclient isf-oauth
oc delete consolelink ibm-spectrum-fusion
```

```
oc delete sppmanagers sppmanager
oc delete updatemanager version

ISF_CRDS=$(oc get crd -o json | jq -r '.items[] | select(.spec.group | endswith("isf.ibm.com")) | .metadata.name')

for CRD in $ISF_CRDS
do
    oc delete crd $CRD --wait=false
    RES_LIST=$( oc get $CRD --all-namespaces -o=jsonpath='{range .items[]}{.metadata.namespace}:{.metadata.name}{"\n"}
{end}')
    sleep 1
    if [ -n "$RES_LIST" ]
    then
        while IFS= read -r res; do
            namespace=$(echo "$res" | cut -d ':' -f 1)
            res_name=$(echo "$res" | cut -d ':' -f 2)
            echo "patching resource $res_name in namespace $namespace"
            oc patch $CRD $res_name -n $namespace --type merge -p '{"metadata":{"finalizers": []}}'
        done <<< "$RES_LIST"
    fi
done

oc delete crd callhomes.scale.spectrum.ibm.com
oc delete crd clusters.scale.spectrum.ibm.com
oc delete crd compressionjobs.scale.spectrum.ibm.com
oc delete crd csiscaleoperators.csi.ibm.com
oc delete crd daemons.scale.spectrum.ibm.com
oc delete crd encryptionconfigs.scale.spectrum.ibm.com
oc delete crd filesystems.scale.spectrum.ibm.com
oc delete crd grafanabridges.scale.spectrum.ibm.com
oc delete crd guis.scale.spectrum.ibm.com
oc delete crd ibmsppcs.sppc.ibm.com
oc delete crd ibmspps.ocp.spp.ibm.com
oc delete crd localdisks.scale.spectrum.ibm.com
oc delete crd pmcollectors.scale.spectrum.ibm.com
oc delete crd recoverygroups.scale.spectrum.ibm.com
oc delete crd remoteclusters.scale.spectrum.ibm.com
oc delete crd cloudcsidisks.scale.spectrum.ibm.com
oc delete crd diskjobs.scale.spectrum.ibm.com
oc delete crd dnss.scale.spectrum.ibm.com
oc delete crd dnsconfigs.scale.spectrum.ibm.com
oc delete crd jobs.scale.spectrum.ibm.com
oc delete crd restripefsjobs.scale.spectrum.ibm.com
oc delete crd stretchclusters.scale.spectrum.ibm.com
oc delete crd stretchclusterinitnodes.scale.spectrum.ibm.com
oc delete crd stretchclustertiebreakers.scale.spectrum.ibm.com
oc delete crd upgradeapprovals.scale.spectrum.ibm.com

log "Scaling application operator back up"
if oc get deployment isf-application-operator-controller-manager -n "$ISF_NS" > /dev/null 2>&1; then
    oc scale deployment isf-application-operator-controller-manager --replicas=1 -n "$ISF_NS"
fi

oc delete OperatorGroup isf-fusionbase -n $FUSION_NS
oc get CatalogSource -n openshift-marketplace -l app=ibm-fusion-hcp -o name | xargs -n1 oc delete -n openshift-marketplace
oc delete ns baas
oc delete ns ibm-spectrum-protect-plus-ns
oc delete ns ibm-spectrum-scale-csi
oc delete ns ibm-spectrum-scale-operator
oc delete ns ibm-spectrum-scale-dns
oc delete ns ibm-spectrum-scale

oc delete subscription ibm-usage-metering-operator -n ums-fusion

oc delete csv -n ums-fusion -l operators.coreos.com/ibm-usage-metering-operator.ums-fusion

oc delete operatorgroup ums-fusion-og -n ums-fusion

oc delete catalogsource ibm-usage-metering-catalog-source -n openshift-marketplace

oc delete ibmusagemetering ibm-usage-metering-fusion -n $FUSION_NS

oc delete ns ums-fusion

oc delete clusterrolebinding isf-sds-serviceability-rolebinding
oc delete clusterrolebinding isf-serviceability-operator-manager-rolebinding
oc delete clusterrolebinding isf-sds-backuprestore-rolebinding
oc delete clusterrolebinding isf-bkprstr-operator-manager-rolebinding
oc delete clusterrolebinding fusionmanager-addon
oc delete clusterrolebinding fusion-support-rolebinding

oc delete role fusionmanager-addon

oc delete ns $FUSION_NS

exit 0
```

Installation and upgrade issues

Use the troubleshooting tips and tricks in IBM Fusion installation and upgrade.

ROSA HCP fresh deployment is not available for IBM Fusion 2.11

Problem statement
: Fresh installation of IBM Fusion 2.11 + Fusion Data Foundation 4.18 and 4.19 on ROSA 4.19 fails due to the failure of Prometheus queries in the console.

Cause
: The UI Storage System creation workflow depends on a node exporter metric. In ROSA clusters, the UI expects a Prometheus instance that does not exist, and hence the workflow fails with a **`502 Bad Gateway`** error.

Resolution
: The resolution is to upgrade to the upcoming fix from Fusion Data Foundation after it becomes available.

Update operator OOMKilled error

Problem statement
: The pods go into crash loop state with the OOMKilled error after OpenShift® Container Platform upgrade.

Resolution
: To resolve the OOMKilled issue for the update operator, do the following resolution steps:

1. Go to IBM Fusion **`clusterserviceversion`** object (Operators > Installed Operators > IBM Fusion operator > YAML tab).
2. Search for the deployment name of the **`isf-update-operator`** (**`isf-update-operator-controller-manager`**) from the list of deployments in the **`clusterserviceversion`** object under **`spec.install.spec.deployments`**.
3. In the specified deployment object, search for the container name manager under the **`spec.template.spec.containers`** and increase the memory limit in the **`resources.limits.memory`**.
4. After changing the limits in the IBM Fusion **`clusterserviceversion`**, the update operator pod restarts with the new limits.
5. If the OOMKilled issue still persists, then follow the steps 1 - 4 again.

x509: certificate signed by an unknown authority

Problem statement
: The x509: certificate signed by an unknown authority error can occur when you trigger a service or firmware upgrade. A sample error is as follows:

```
Internal error occurred: failed calling webhook "mupdatemanager.kb.io": failed to call webhook: Post "https://isf-update-opera
```

Resolution
: Do the following resolution steps:

1. In the OpenShift Container Platform console, go to Home > Search.
2. From the **`Resources`** drop-down list, select `MutatingWebhookConfiguration`.
3. Select the **`Label`** drop-down list and change it to **`Name`**.
4. Search for **`mupdatemanager`**. Check whether there are more than one instance of **`mupdatemanager.*`** webhook. If so, take a backup of the older one and delete it.
5. Go back to Home > Search page.
6. From the Resources, select **`ValidatingWebhookConfiguration`**.
7. Search for **`vupdatemanager`**. Check whether there are more than one instance of **`vupdatemanager.*`** webhook. If so, take the backup of the older one and delete it.

Known issues

If the IBM Fusion installation is stuck in the **`custom-ns`** namespace with the error **`deployment of isf-serviceability-operator not ready before timeout`**, ignore it and continue with the installation.

IBM Fusion Access for SAN

IBM Fusion Access for SAN is a cloud-native solution that helps enterprises transition from traditional virtualization environments to OpenShift® while reusing existing Storage Area Network (SAN) infrastructure.

IBM Fusion Access for SAN is a solution that provides a scalable clustered file system for enterprise storage, primarily designed to offer access to consolidated, block-level data storage. It presents storage devices, such as disk arrays, to the operating system as if they were direct-attached storage.

This solution supports toward enterprise storage for Red Hat OpenShift Virtualization and uses existing SAN infrastructure. A SAN is a dedicated network of storage devices that is not accessible through the local area network (LAN).

IBM Fusion Access for SAN offers the following key capabilities and benefits:

Simplified user experience
: IBM Fusion Access for SAN brings a wizard-driven user interface that simplifies the installation and configuration of storage clusters, file systems, and storage classes.

Use existing infrastructure
: You can use your existing SAN investments, including Fibre Channel (FC) and iSCSI technologies, as you transition to or expand with Red Hat OpenShift Virtualization.

Scalability
: The storage cluster is designed to scale with OpenShift Container Platform clusters and virtual machine workloads. It can support up to approximately 3000 VMs on 6 bare-metal hosts, with possibilities for further scaling by adding more file systems or using specific storage class parameters.

Consolidated and shared storage
: SANs enable multiple servers to access a large, shared data storage capacity. This architecture facilitates automatic data backup and monitors the storage and backup processes continuously.

High-speed data transfer
: By using a dedicated high-speed network for storage, IBM Fusion Access for SAN overcomes the data transfer bottlenecks that can occur over a traditional LAN, especially for large volumes of data.

File-level access
: Although a SAN primarily operates at the block level, file systems that are built on top of SAN storage can provide file-level access through shared-disk file systems.

Centralized management
: The underlying SAN software manages servers, storage devices, and the network to ensure that data moves directly between storage devices with minimal server intervention. It also supports centralized management and configuration of SAN components like Logical Unit Numbers (LUNs).

For more information about IBM Fusion Access for SAN, see the FAQ section of IBM Fusion Access for SAN.

To deploy IBM Fusion Access for SAN, see Deploying IBM Fusion Access for SAN.

- **Deploying IBM Fusion Access for SAN**
  Describes the supported methods for deploying IBM Fusion Access for Storage Area Network (SAN) in IBM Fusion environments.
- **IBM Fusion Access for SAN using Secure Boot**
  Secure Boot uses Unified Extensible Firmware Interface (UEFI) technology to verify the integrity of the boot process. The firmware validates each component in the boot chain and allows execution only when a trusted key signs the code. When the system detects tampering or an unsigned component, it stops the boot process.

Deploying IBM Fusion Access for SAN

Describes the supported methods for deploying IBM Fusion Access for Storage Area Network (SAN) in IBM Fusion environments.

Before you start

The deployment method for IBM Fusion Access for SAN depends on the installed version of Red Hat® OpenShift® Container Platform in your environment.

- For Red Hat OpenShift Container Platform 4.21 or later, you can deploy IBM Fusion Access for SAN either by configuring it in Fusion Data Foundation as an external system or by installing the IBM Fusion Access for SAN operator.
- For Red Hat OpenShift Container Platform 4.20 or earlier, you can deploy IBM Fusion Access for SAN only by installing the IBM Fusion Access for SAN operator.

For more information on the supported deployment methods and their limitations, see the following table:

Table 1. IBM Fusion Access for SAN deployment methods

Red Hat OpenShift Container Platform version	Deployment method	Key details and limitations
4.21 or later	Configure IBM Fusion Access for SAN in Fusion Data Foundation as an external system. For instructions, see Configuring as an external system in Fusion Data Foundation.	• Requires Fusion Data Foundation 4.21 or later. • Installs Global Data Platform or IBM Storage Scale (CNSA) version 6.0.1.0. • Global Data Platform and Content-Aware Storage services cannot coexist with IBM Fusion Access for SAN because they rely on the same underlying IBM Storage Scale Container Storage Interface (CSI) driver resources.
	Install IBM Fusion Access for SAN using the IBM Fusion Access for SAN operator. For instructions, see Configuring as an IBM Fusion Access for SAN operator.	• Deploys IBM Fusion Access for SAN as a standalone operator. • Global Data Platform and Content-Aware Storage services cannot coexist with IBM Fusion Access for SAN because they rely on the same underlying IBM Storage Scale Container Storage Interface (CSI) driver resources.
4.20 or earlier	Install IBM Fusion Access for SAN using the IBM Fusion Access for SAN operator. For instructions, see Configuring as an IBM Fusion Access for SAN operator.	• Deploys IBM Fusion Access for SAN as a standalone operator. • Global Data Platform and Content-Aware Storage services cannot coexist with IBM Fusion Access for SAN because they rely on the same underlying IBM Storage Scale Container Storage Interface (CSI) driver resources.

Additional information

- IBM Fusion Access for SAN supports both Secure Boot-enabled and non-Secure Boot-enabled clusters. To enable Secure Boot in an IBM Fusion Access for SAN cluster, see IBM Fusion Access for SAN using Secure Boot.
- You can deploy IBM Fusion Access for SAN in a disconnected (air-gapped) OpenShift environment. For instructions, see Deploying in a disconnected (air-gapped) OpenShift environment.

Configuring as an external system in Fusion Data Foundation

Prerequisites
Verify that your environment meets the following prerequisites:

- Worker nodes with attached SAN storage.
- An OpenShift container registry with persistent storage. For configuration instructions, see Configuring the registry for bare metal in *Red Hat OpenShift Container Platform* documentation.
- Worker nodes that are connected to the same LUNs.
  Note: A shared LUN is a shared disk that is accessed by all worker nodes simultaneously.
- Global Data Platform and Content-Aware Storage services are not installed.

For information about hardware requirements for deploying IBM Fusion Access for SAN, see Hardware requirements.

Limitations

- Limitations for IBM Fusion Access for SAN are based on the IBM Storage Scale Container Native limitations. For more information, see IBM Storage Scale container native documentation.
- Hosted Control Plane clusters are not supported.
- IBM Fusion Access for SAN is not supported on IBM Z (s390x) and IBM LinuxONE platforms.

Procedure

1. Install the Fusion Data Foundation version 4.21 or later by following the instructions mentioned in Installing Fusion Data Foundation.
2. Configure the IBM Fusion Access for SAN in Fusion Data Foundation as an external system by following the instructions mentioned in Configuring with IBM Fusion Access for Storage Area Network.

Configuring as an IBM Fusion Access for SAN operator

Prerequisites
Verify that your environment meets the following prerequisites:

- Bare metal worker nodes with attached SAN storage.
- An OpenShift container registry with persistent storage. For configuration instructions, see Configuring the registry for bare metal in *Red Hat OpenShift Container Platform* documentation.
- Worker nodes that are connected to the same LUNs
  Note: A shared LUN is a shared disk that is accessed by all worker nodes simultaneously.
- Global Data Platform and Content-Aware Storage services are not installed.

For information about hardware requirements for deploying IBM Fusion Access for SAN, see Hardware requirements.

Limitations

- Limitations for IBM Fusion Access for SAN are based on the IBM Storage Scale Container Native limitations. For more information, see IBM Storage Scale container native documentation.
- Hosted Control Plane clusters are not supported.

Procedure

1. Install the IBM Fusion Access for SAN by following the instructions mentioned in Installing IBM Fusion Access for SAN operator.
2. Create a Kubernetes pull secret by following the instructions mentioned in Creating a Kubernetes pull secret.
3. Create the `FusionAccess` custom resource (CR) by following the instructions mentioned in Creating the FusionAccess custom resource (CR).
4. Create a storage cluster by following the instructions mentioned in Creating a storage cluster.
5. Creating a file system by following the instructions mentioned in Creating a filesystem.

- **Installing IBM Fusion Access for SAN operator**
  Use this procedure to install IBM Fusion Access for Storage Area Network (SAN) operator.
- **Creating a Kubernetes pull secret**
  Use the instructions to create a Kubernetes pull secret.
- **Creating the FusionAccess custom resource (CR)**
  Use the instructions to create a `FusionAccess` CR.
- **Creating a storage cluster**
  Use this procedure to create a storage cluster.
- **Creating a filesystem**
  Use this procedure to create a filesystem.
- **Deploying in a disconnected (air-gapped) OpenShift environment**
  The following section describes the end-to-end process to deploy IBM Fusion Access for Storage Area Network (SAN) in a disconnected (air-gapped) environment. It includes registry mirroring, operator installation, configuration, and validation steps to ensure that all components are properly configured for air-gapped deployments.

Installing IBM Fusion Access for SAN operator

Use this procedure to install IBM Fusion Access for Storage Area Network (SAN) operator.

Before you begin

Ensure that you have the following prerequisites:

1. Access to the cluster as a user with the `cluster-admin` role.
2. A working container registry that is enabled.

About this task

Install the IBM Fusion Access for SAN operator from the OperatorHub in the OpenShift® Container Platform web console.

Procedure

1. Log in to the OpenShift Container Platform web console.
2. Go to Operators > OperatorsHub.

3. In the Filter by keyword field, type **Fusion Access for SAN**.
4. Select the Fusion Access for SAN tile and click Install.
5. On the Install Operator page, keep the default selections for Update Channel, Version, and Installation mode.
6. Verify that Operator recommended Namespace is selected for the Installed Namespace.
   This installs the operator in the **ibm-fusion-access** namespace. If this namespace does not exist, it is automatically created.
   Warning: If you install the IBM Fusion Access for SAN operator in a namespace other than **ibm-fusion-access**, then IBM Fusion Access for SAN does not work.
7. Verify that the Automatic default is selected for Update Approval.
   This enables automatic updates when a new release is available.
8. Click Install to install the operator.
9. Go to Operators > OperatorsHub, and verify that the IBM Fusion Access for SAN operator is displayed.

Creating a Kubernetes pull secret

Use the instructions to create a Kubernetes pull secret.

Before you begin

Ensure that you have the following prerequisites:

1. Install the **oc** CLI.
2. Access to the cluster as a user with the **cluster-admin** role.
3. Install the IBM Fusion Access for SAN operator and created the **ibm-fusion-access** namespace in the process. For more information about the procedure, see Installing IBM Fusion Access for SAN operator.

About this task

Create a Kubernetes secret object to hold the IBM entitlement key to pull the required container images from the IBM container registry.

Procedure

1. Log in to the **IBM Container software library** with your IBM Fusion Access for SAN IBMid and password.
2. In the IBM Container software library, get the entitlement key:
   a. If you do not have an entitlement key yet, click Get entitlement key or Add new key, and then click Copy.
   b. If you already have an entitlement key, click Copy.
3. Save the entitlement key.
4. Create the secret object by running the **oc create** command:

```
$ oc create secret -n ibm-fusion-access generic fusion-pullsecret \
--from-literal=ibm-entitlement-key=<ibm-entitlement-key>
```

5. Verify whether the pull secret is created successfully as follows:
   a. Log in to the OpenShift® Container Platform web console.
   b. Go to Workloads > Secrets.
   c. Ensure that **fusion-pullsecret** is present in the list.

Creating the FusionAccess custom resource (CR)

Use the instructions to create a **FusionAccess** CR.

Before you begin

Ensure that you have the following prerequisites:

1. Access to the cluster as a user with the **cluster-admin** role.
2. Install the IBM Fusion Access for SAN operator and created the **ibm-fusion-access** namespace in the process. For more information about the procedure, see Installing IBM Fusion Access for SAN operator.
3. Create a Kubernetes pull secret. For more information about the procedure, see Creating a Kubernetes pull secret.

About this task

Create a **FusionAccess** CR that triggers the installation of the correct version of IBM Storage Scale and detects worker nodes with shared Logical Unit Numbers (LUNs).

Procedure

1. Log in to the OpenShift® Container Platform web console.
2. Go to Operators > Installed Operators.
3. Click the IBM Fusion Access for SAN operator you installed.
4. In the Fusion Access for SAN page, select the Fusion Access tab.
5. Click Create FusionAccess.
6. On the Create FusionAccess page, enter the object Name.
7. Optional: You can choose to add Labels if they are relevant.

8. Select the IBM Storage Scale from the drop-down list.
   This version is the same version that used to be installed in the IBM Fusion Access for SAN operator.
9. Click Create.
10. Verify whether the `FusionAccess` CR is created successfully.
    a. Go to Fusion Access tab in the Fusion Access for SAN operator page.
    b. Check that the created `FusionAccess` CR appears with the status Ready.

Creating a storage cluster

Use this procedure to create a storage cluster.

Before you begin

Ensure that you have the following prerequisites:

1. Bare Metal worker nodes with visible and attached shared Logical Unit Numbers (LUNs).
2. Install the IBM Fusion Access for SAN operator. For more information about the procedure, see Installing IBM Fusion Access for SAN operator.
3. Create the `FusionAccess` CR in the `ibm-fusion-access` namespace. For more information about the procedure, see Creating the FusionAccess custom resource (CR).

About this task

After you installed the IBM Fusion Access for SAN operator, you can create a storage cluster with shared storage nodes. The wizard to create the storage cluster in the OpenShift® Container Platform web console provides step-by-step instructions and lists the relevant worker nodes with shared disks.

Procedure

1. Log in to the OpenShift Container Platform web console.
2. Go to Storage > Fusion Access for SAN.
3. Click Create storage cluster.
4. Select the worker nodes that have shared LUNs.
   Note: You can select only worker nodes with a minimum of 20 GB of RAM from the list.
5. Click Create storage cluster.
   The page reloads, opening the `Fusion Access for SAN` page for the new storage cluster.

Creating a filesystem

Use this procedure to create a filesystem.

Before you begin

- Ensure that you created the IBM Fusion Access for SAN cluster.

About this task

You need to create a file system to represent your required storage. The file system is based on the storage available in the worker nodes that you selected when creating the storage cluster.

Procedure

1. Log in to the OpenShift® Container Platform web console.
2. Go to Storage > Fusion Access for SAN.
3. In the File systems tab, click Create file system.
4. Enter a Name for the new file system.
5. Select the Logical Unit Numbers (LUNs) that you want to use as the storage volumes for your file system.
6. Click Create file system.
   The Fusion Access for SAN page reloads, and the new file system appears in the File systems tab.
7. Repeat this procedure for each filesystem that you want to create.
8. Verify whether the file system is created successfully.
   a. Observe the Status of the file system in the File systems tab until it is marked as Healthy.
   b. Click the StorageClass for the file system.
   c. In the YAML tab, verify the following values:
      i. The value in the `name` field is the name of the file system that you created.
      ii. The value in the `provisioner` field is `spectrumscale.csi.ibm.com`.
      iii. The value in the `volBackendFs` field matches the name of the file system that you created.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: filesystem1
  uid: eb410309-a043-a89b-9bb05483872a
  resourceVersion: '87746'
```

```
  creationTimestamp: '2025-05-14T12:30:08Z'
  managedFields:
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: filesystem1
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

What to do next

1. After you create a storage cluster with file systems, you can create a virtual machine (VM) on the storage cluster.
2. Create a VM from an instance type or template and select one of the file systems you created as the storage type.
   - Creating virtual machines from instance types
   - Creating virtual machines from templates

Deploying in a disconnected (air-gapped) OpenShift environment

The following section describes the end-to-end process to deploy IBM Fusion Access for Storage Area Network (SAN) in a disconnected (air-gapped) environment. It includes registry mirroring, operator installation, configuration, and validation steps to ensure that all components are properly configured for air-gapped deployments.

Before you begin

Red Hat® OpenShift® Container Platform 4.21 or later

IBM Fusion Access for SAN is integrated with Fusion Data Foundation. To deploy IBM Fusion Access for SAN in a disconnected (air-gapped) environment, complete the following procedures:

1. Mirror the Red Hat operator images as described in Mirroring Red Hat operator images to enterprise registry.
2. Mirror the Fusion Data Foundation images as described in Mirroring Fusion Data Foundation images.
3. Mirror the IBM Storage Scale 6.0.1.0 images as described in Mirroring IBM Storage Scale 6.0.1.0 images.

Red Hat OpenShift Container Platform 4.20 or earlier

Use this procedure to deploy IBM Fusion Access for SAN in a disconnected (air-gapped) environment.

Ensure that the following prerequisites are met:

- Red Hat OpenShift Container Platform version 4.20 or earlier.
- IBM Fusion Access for SAN Software version 1.1.0 or later.
- Worker nodes with SAN-attached storage (Fibre Channel or iSCSI).
- Access to a connected environment for image mirroring.
- IBM Cloud entitlement key for image registry authentication.
- Configured internal (local) container image registry accessible from all cluster nodes.
- Adequate CPU, memory, and disk resources for Fusion Access components.
- Network connectivity between OpenShift cluster nodes.

About this task

IBM Fusion Access for SAN extends IBM Fusion Software's storage capabilities by providing high-performance, low-latency SAN-based storage access for Red Hat OpenShift Virtualization environments.

Environment architecture

The reference architecture for deploying IBM Fusion Access for SAN in a disconnected environment includes:

- One OpenShift 4.20.8 cluster (control plane and worker nodes).
- One external private registry (for example, `registry.xxx.xxx:5000`) for mirrored images.
- Worker nodes with SAN-attached storage (Fibre Channel or iSCSI).

Procedure

1. Prepare the disconnected environment. Disconnected deployments require mirrored container images from external sources into a local image registry.
   a. Whitelist the following URLs to download the images on the mirror host:
      - `registry.access.redhat.com`
      - `quay.io`
      - `registry.redhat.io`
      - `cp.icr.io`
      - `icr.io`
   b. Prepare the `ImageSetConfiguration` file to mirror OpenShift platform, operators, and IBM images into the disconnected registry.
   Example configuration:

```
# cat fusion-access-san-mirror-ubi9.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
mirror:
  platform:
    graph: true
    channels:
    - name: candidate-4.20
      type: ocp
      minVersion: 4.20.0
```

```
      maxVersion: 4.20.8
  operators:
  - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.20
    packages:
      - name: kubevirt-hyperconverged
      - name: kernel-module-management
  - catalog: registry.redhat.io/redhat/certified-operator-index:v4.20
    packages:
      - name: openshift-fusion-access-operator
  additionalImages:
  - name: cp.icr.io/cp/gpfs/csi/csi-attacher@sha256:69888dba58159c8bc0d7c092b9fb97900c9ca8710d088b0b7ea7bd9052df86f6
  - name: cp.icr.io/cp/gpfs/csi/csi-node-driver-
registrar@sha256:d7138bcc3aa5f267403d45ad4292c95397e421ea17a0035888850f424c7de25d
  - name: cp.icr.io/cp/gpfs/csi/csi-
provisioner@sha256:d5e46da8aff7d73d6f00c761dae94472bcda6e78f4f17b3802dc89d44de0111b
  - name: cp.icr.io/cp/gpfs/csi/csi-resizer@sha256:8ddd178ba5d08973f1607f9b84619b58320948de494b31c9d7cd5375b316d6d4
  - name: cp.icr.io/cp/gpfs/csi/csi-
snapshotter@sha256:5f4bb469fec51147ce157329dab598c758da1b018bad6dad26f0ff469326d769
  - name: cp.icr.io/cp/gpfs/csi/ibm-spectrum-scale-csi-
driver@sha256:711f9fc45969639da712b0823139549cdbedead0be9453c4243e1a873e7737f1
  - name: cp.icr.io/cp/gpfs/csi/livenessprobe@sha256:2c5f9dc4ea5ac5509d93c664ae7982d4ecdec40ca7b0638c24e5b16243b8360f
  - name: cp.icr.io/cp/gpfs/data-management/ibm-spectrum-scale-
daemon@sha256:07ae36c1a8539aee091cff19261aa14a9d220e6994132a14d2d8c537d596f9ec
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
coredns@sha256:6a317b23fee629c0b07eb95d34ab7593bb38d41bffc5d1cd4cb2870539c66cd4
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-core-
init@sha256:51dc287dd9ae2f8dcb60c1678fe8b535bb72e29faad24108d55b7cfe62362777
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-grafana-
bridge@sha256:b3b91b2f6729ecb3544b6b9fc539e2abbe425dfe497ce88deba7deed55305c7c
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
gui@sha256:57ea64cdd612aef7a5f01dfbd41677092c0f565c35ef7e5394e50799dc3796e1
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
logs@sha256:3b48c09f3641c10c63378a2dda806673fe7ea9c43774112f67bd8f8bd4e53b93
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
monitor@sha256:c77c0b3c6f7373136b581dabf2f57899c8a742d1cb4501d739af95cba151a438
  - name: cp.icr.io/cp/gpfs/ibm-spectrum-scale-
pmcollector@sha256:ad1e25622e325f4aa3000e680bd747bea50a22b65a8472cbc20640b730c86fa4
  - name: cp.icr.io/cp/gpfs/postgres@sha256:0bcc5bbbb2aa9c9b4c6505845918c7eb55d783cf5c1f434fac33012579fb149d
  - name: icr.io/cpopen/ibm-spectrum-scale-csi-
operator@sha256:2c2e9c630a45da80321bbf1b63e9152e0cb7cc14390ee3879fd736e0a6974464
  - name: icr.io/cpopen/ibm-spectrum-scale-must-
gather@sha256:798559d9dfabfbd2cca1224fde8f6b0cbc4c96fd9d01212ea07bfdb4e40c8818
  - name: icr.io/cpopen/ibm-spectrum-scale-
operator@sha256:6dcb6ca5430deea044cfd029df4a9f0576586e824af07d9c61f4630ee4798e11
  - name: registry.redhat.io/ubi9/ubi:latest
  - name: registry.redhat.io/ubi9/ubi-minimal:latest
  - name: registry.redhat.io/ubi9/ubi-micro:latest
```

c. Perform a dry run of mirroring and ensure that all images are included in the `mapping.txt` file:

```
oc mirror --config fusion-access-san-mirror.yaml file://mirror-output --dry-run
```

d. Download and push the images to the private registry:

```
oc mirror --config fusion-access-san-mirror.yaml docker://registry.xxx.xxx:5000
```

2. Configure OpenShift to use local registry. After image mirroring, create `ImageContentSourcePolicy` and mirror sets to redirect OpenShift image pulls to the local registry.
`ImageContentSourcePolicy` example:

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: combined-policy
  labels:
    operators.openshift.org/catalog: "true"
spec:
  repositoryDigestMirrors:
    - mirrors:
        - registry.xxx.xxx:5000/container-native-virtualization
      source: registry.redhat.io/container-native-virtualization
    - mirrors:
        - registry.xxx.xxx:5000/openshift4
      source: registry.redhat.io/openshift4
    - mirrors:
        - registry.xxx.xxx:5000/kmm
      source: registry.redhat.io/kmm
    - mirrors:
        - registry.xxx.xxx:5000/ibm
      source: registry.connect.redhat.com/ibm
    - mirrors:
        - registry.xxx.xxx:5000/cpopen
      source: icr.io/cpopen
    - mirrors:
        - registry.xxx.xxx:5000/migration-toolkit-virtualization
      source: registry.redhat.io/migration-toolkit-virtualization
    - mirrors:
        - registry.xxx.xxx:5000/ocp4/openshift4
      source: quay.io/openshift-release-dev/ocp-release
    - mirrors:
        - registry.xxx.xxx:5000/ocp4/openshift4
      source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
    - mirrors:
        - registry.xxx.xxx:5000/cp
      source: cp.icr.io/cp
```

```
    - mirrors:
        - registry.xxx.xxx:5000/ubi8
      source: registry.access.redhat.com/ubi8

apiVersion: config.openshift.io/v1
kind: ImageTagMirrorSet
metadata:
  name: redhat-registry-tag-mirror
spec:
  imageTagMirrors:
    - mirrorSourcePolicy: NeverContactSource
      mirrors:
        - registry.xxx.xxx:5000/ubi9/ubi-minimal
      source: registry.redhat.io/ubi9/ubi-minimal
    - mirrorSourcePolicy: NeverContactSource
      mirrors:
        - registry.xxx.xxx:5000/ubi9/ubi
      source: registry.redhat.io/ubi9/ubi

apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: redhat-registry-mirror
spec:
  imageDigestMirrors:
    - mirrorSourcePolicy: AllowContactingSource
      mirrors:
        - registry.xxx.xxx:5000
      source: registry.redhat.io
    - mirrorSourcePolicy: AllowContactingSource
      mirrors:
        - registry.xxx.xxx:5000
      source: registry.access.redhat.com
```

Following are the final set of images, which require to install IBM Fusion Access for SAN components in OpenShift cluster:

```
{"name":"cp/gpfs/csi/ibm-spectrum-scale-csi-driver","tags":["latest"]}
{"name":"cp/gpfs/data-access/ibm-spectrum-scale-daemon","tags":["latest"]}
{"name":"cp/gpfs/data-management/ibm-spectrum-scale-daemon","tags":["latest"]}
{"name":"cp/gpfs/erasure-code/ibm-spectrum-scale-daemon","tags":["latest"]}
{"name":"cp/gpfs/ibm-spectrum-scale-core-init","tags":["latest"]}
{"name":"cp/gpfs/ibm-spectrum-scale-coredns","tags":["latest"]}
{"name":"cp/gpfs/ibm-spectrum-scale-grafana-bridge","tags":["latest"]}
{"name":"cp/gpfs/ibm-spectrum-scale-gui","tags":["latest"]}
{"name":"cp/gpfs/ibm-spectrum-scale-logs","tags":["latest"]}
{"name":"cp/gpfs/ibm-spectrum-scale-monitor","tags":["latest"]}
{"name":"cp/gpfs/ibm-spectrum-scale-pmcollector","tags":["latest"]}
{"name":"cpopen/fusion-access/console-plugin-rhel9","tags":["30603e12"]}
{"name":"cpopen/fusion-access/devicefinder-rhel9","tags":["9ac58bbc"]}
{"name":"cpopen/fusion-access-controller-rhel9-operator","tags":["befa1c1"]}
{"name":"cpopen/ibm-spectrum-scale-csi-operator","tags":["latest"]}
{"name":"cpopen/ibm-spectrum-scale-must-gather","tags":["latest"]}
{"name":"cpopen/ibm-spectrum-scale-operator","tags":["latest"]}
{"name":"ibm/fusion-access-bundle","tags":["82382647"]}
{"name":"kmm/kernel-module-management-must-gather-rhel9","tags":["b0407df0","1edc9aa8","75ae837d","8f5c05af","1f055e3b"]}
{"name":"kmm/kernel-module-management-operator-bundle","tags":["c5613138","d7699aa4","d995ef03","cc267b84","4f892fcd"]}
{"name":"kmm/kernel-module-management-rhel9-operator","tags":["73c13de7","bbd482d3","773c26c0","f05eb9ff","80947f31"]}
{"name":"kmm/kernel-module-management-signing-rhel9","tags":["953052b5","7512eb99","60711d48","779023b6","6bd59035"]}
{"name":"kmm/kernel-module-management-webhook-server-rhel9","tags":["4442c898","6d768cac","871b87da","a9e34cf5"]}
{"name":"kmm/kernel-module-management-worker-rhel9","tags":["56032742","e4f4196d","ca89d35f","d96f02c1","b81dc29a"]}
registry.redhat.io/ubi9/ubi@sha256:dec374e05cc13ebbc0975c9f521f3db6942d27f8ccdf06b180160490eef8bdbc
registry.access.redhat.com/ubi8/ubi-micro@sha256:37552f11d3b39b3360f7be7c13f6a617e468f39be915cd4f8c8a8531ffc9d43d
registry.redhat.io/ubi9/ubi-minimal@sha256:34880b64c07f28f64d95737f82f891516de9a3b43583f39970f7bf8e4cfa48b7
registry.redhat.io/ubi9/ubi-micro@sha256:990002083442f6a93cd3249da32ecb7c3f6be778a1bec3a73a9c17fbc40edc15
```

3. Install the IBM Fusion Access for SAN operator to enable Red Hat OpenShift Virtualization with IBM Fusion Access for SAN. For more information, see Installing IBM Fusion Access for SAN operator.
    a. In the OpenShift console, go to Operators > OperatorHub, and install the IBM Fusion Access for SAN operator.
    b. Verify that the IBM Fusion Access for SAN and Kernel Module Management (KMM) operators show status Succeeded.
    c. Create a Kubernetes pull secret and generate an entitlement key. For more information, see Creating a Kubernetes pull secret.
    Note: By default, the KMM operator uploads GPFS module images to the integrated internal OpenShift image registry. Ensure that this image registry is not configured with `emptyDir` storage. Use a ReadWriteMany (RWX) PVC or S3-compatible backend.
4. Configure OpenShift internal image registry for KMM operator usage. For more information, see the Red Hat documentation.
5. If the OpenShift internal registry is not preferred, optionally, you can configure KMM operator to use a different external image registry. Create a ConfigMap and secret after installing the Fusion Access for SAN operator.
Note: This step must be performed before creating the `FusionAccess` object CR.

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: kmm-image-config
  namespace: ibm-fusion-access
data:
  kmm_image_registry_url: "<registry-url>"
  kmm_image_repo: "<repo>"
  kmm_tls_insecure: "false"
  kmm_tls_skip_verify: "false"
  kmm_image_registry_secret_name: "<kmm-push-secret>"
---
apiVersion: v1
kind: Secret
metadata:
  name: "<kmm-push-secret>"
```

```
  namespace: ibm-fusion-access
data:
  .dockerconfigjson: "<secret-data>"
type: kubernetes.io/dockerconfigjson
```

6. Create a `FusionAccess` custom resource (CR) for SAN instance:
   a. In the OpenShift console, go to Operators > Installed Operators and select Fusion Access for SAN.
   b. Create a `FusionAccess` CR.
   c. Verify that the CR status shows Ready.
   d. Ensure that all KMM instances are created and the `gpfs-module` YAML section status shows Success.
   e. Confirm that all pods in the `ibm-fusion-access` namespace are in Running and Ready state.
7. Create the storage cluster:
   a. In the OpenShift console, go to Storage > Fusion Access for SAN.
   b. Select worker nodes with shared SAN disks.
   c. Monitor installation progress in the `ibm-spectrum-scale` namespace.
   d. Verify that all pods are in Running and Ready state.
8. Create the filesystem:
   a. Select disks and specify a filesystem name.
   b. Monitor filesystem creation in the `ibm-fusion-access` and `ibm-spectrum-scale` namespaces.
   c. In the OpenShift console, go to Storage > StorageClasses and verify whether the `StorageClass` is created and ready.
   d. Go to Storage > PersistentVolumeClaims and create a RWX PVC using the Fusion Access for SAN `StorageClass`.
9. Optionally, access the Scale GUI using the URL provided in the File System page to review configuration and fileset details.

IBM Fusion Access for SAN using Secure Boot

Secure Boot uses Unified Extensible Firmware Interface (UEFI) technology to verify the integrity of the boot process. The firmware validates each component in the boot chain and allows execution only when a trusted key signs the code. When the system detects tampering or an unsigned component, it stops the boot process.

About this task

Secure Boot blocks unsigned kernel modules. To load custom kernels or kernel modules, you must create a signing key and enroll the corresponding public certificate in the Machine Owner Key (MOK) database. After enrollment, the system trusts kernel modules that are signed with the private key. For more information, see Create an X.509 public and private key pair to sign custom kernel modules.

The following procedure enables Secure Boot, prepares signing material, enrolls trust keys on target nodes, and validates the configuration.

Procedure

1. Run the following commands to verify whether Secure Boot is enabled on the nodes:

   ```
   oc debug node/<node>
   ```

   ```
   chroot /host
   ```

   ```
   mokutil --sb-state
   ```

   Note: All worker nodes in the cluster must be enabled with Secure Boot.
2. Create an X.509 public and private key pair to sign kernel modules.
   Note: Run the following commands on a RHEL workstation once for every OpenShift® cluster and generate the key pair.
   a. Install the required tools:

      ```
      sudo dnf install -y pesign mokutil openssl
      ```

   b. Generate a self-signed key pair:

      ```
      sudo efikeygen --dbdir /etc/pki/pesign --self-sign --module --common-name "CN=KMM SecureBoot Key,O=MyOrg" --nickname
      "kmm_sb_key"
      ```

   c. Verify that the key is created:

      ```
      sudo certutil -L -d /etc/pki/pesign
      ```

      The output must include `kmm_sb_key`.

   d. Export the certificate in PEM format:

      ```
      certutil -d /etc/pki/pesign -n 'kmm_sb_key' -Lr > sb_cert.cer
      ```

   e. Export the private key in PKCS#12 format:

      ```
      pk12util -o sb_cert.p12 -n 'kmm_sb_key' -d /etc/pki/pesign
      ```

   f. Extract the private key in PEM format:

      ```
      openssl pkcs12 -in sb_cert.p12 -out sb_cert.priv -nocerts -nodes
      ```

   g. Convert the PEM certificate to DER format for MOK import:

      ```
      openssl x509 -in sb_cert.cer -outform DER -out sb_cert.der
      ```

3. Copy the DER certificate on the nodes where Secure Boot is enabled.

   ```
   scp sb_cert.der core@worker0.ocp1.vmlocal:/home/core/
   ```

4. Import the certificate that uses MOK utility.
   a. Open a debug session to the node:

      ```
      oc debug node/<node>
      ```

   b. Change the root directory to the host file system:

      ```
      chroot /host
      ```

   c. Import the certificate:

      ```
      sudo mokutil --import /home/core/sb_cert.der
      ```

   d. Enter a password when prompted.
      Remember: Save this password for use during key enrollment.
   e. Restart the node:

      ```
      systemctl reboot
      ```

      Note: You must have console access to the server before the reboot. During the reboot, the BIOS prompts you to manually accept the new key. Accept the key when prompted.
5. Enter the MOK manager and enroll the key.
   Note: Enroll the key on all nodes that are part of IBM Fusion Access.
6. Verify that the system enrolled the MOK successfully after the node starts.
   a. Start a debug session on the node:

      ```
      oc debug node/<node>
      ```

   b. Change the root directory:

      ```
      chroot /host
      ```

   c. List enrolled MOK keys:

      ```
      mokutil --list-enrolled
      ```

   d. Verify the platform key ring:

      ```
      keyctl list %:.platform
      ```

      The output includes the Secure Boot signing key.

Results

The environment runs with UEFI Secure Boot enabled.

What to do next

After the Secure Boot is enabled, follow these steps to use the enrolled signing key and load kernel modules without interruption:

1. Create the `ibm-fusion-access` namespace:

   ```
   oc create ns ibm-fusion-access
   ```

   Note: This command is allowed to fail if the `ibm-fusion-access` namespace already exists.
2. Create the following secrets:
   - Private key

     ```
     oc create secret generic secureboot-signing-key \
       -n ibm-fusion-access \
       --from-file=key=sb_cert.priv
     ```

   - Public key

     ```
     oc create secret generic secureboot-signing-key-pub \
       -n ibm-fusion-access \
       --from-file=cert=sb_cert.der
     ```

3. Create the `FusionAccess` custom resource. For more information, see Creating the FusionAccess custom resource (CR).

Storage

IBM Fusion offers a variety of storage options to meet the demands of high-performance workloads and fulfill the storage needs of virtual machines.

Fusion storage services

Depending on your workload requirements, choose between either of the following services:

- Fusion Data Foundation storage is optimized for container workload management, ensuring efficient orchestration and resource allocation.
- Global Data Platform, on the other hand, is designed for complex AI processing, providing enhanced computational power and scalability.

Fusion Data Foundation

Fusion Data Foundation is designed to provide persistent storage and cluster data management. It supports containerized environments and integrates seamlessly with Red Hat® OpenShift®. It supports file, block, and object storage, and provides a consistent storage experience across hybrid cloud environments. For more information about Fusion Data Foundation service, see Fusion Data Foundation.

Object Storage
: Fusion Data Foundation includes the following object storage:

  - RADOS gateway is deployed in external mode for persistent storage across containerized environments.
  - NooBaa acts as a Multicloud Object Gateway to manage data across multiple storage backend.

Global Data Platform

Global Data Platform is a clustered file system that provides concurrent access to a single file system or set of file systems from multiple nodes. The nodes can be SAN-attached, network-attached, a mixture of SAN-attached and network-attached, or in a shared-nothing cluster configuration. This enables high-performance access to this common set of data to support a scale-out solution or to provide a high availability platform. The IBM Storage Scale storage is designed for building a global data platform for AI, high-performance computing (HPC), advanced analytics, and other demanding workloads. It supports file and object storage for unstructured data. For more information about Global Data Platform service, see Global Data Platform. For more information about IBM Storage Scale service, see IBM Storage Scale documentation.

Object Storage
: Multi-Cloud Gateway (MCG) in Global Data Platform is an on-prem deployment. If the Global Data Platform ECE is installed, then you can see the MCG tile for it.

Global Data Platform Remote Mount

The Global Data Platform remote mount enables you to connect to a remote IBM Storage Scale cluster as if it were part of your local cluster. This is particularly useful for distributed workloads and hybrid cloud environments. For more information about, see Remote IBM Storage Scale storage cluster in IBM Fusion.

IBM Storage Ceph

When you install Fusion Data Foundation, you can connect or configure the Fusion Data Foundation with an external IBM Storage Ceph cluster to use it as a storage provider. As you connect and manage additional storage resources, you can scale your storage infrastructure as needed.

IBM FlashSystem

When you install Fusion Data Foundation, you can connect or configure the Fusion Data Foundation with an external IBM FlashSystem cluster to use it as a storage provider. As you connect and manage additional storage resources, you can scale your storage infrastructure as needed.

Combining these storage options

The different storage installation combinations that you can do using these options, see Storage deployment combinations from the user interface.

- **Storage deployment combinations from the user interface**
  Describes the supported storage deployment options available in the IBM Fusion user interface, including Fusion Data Foundation local, dynamic, and external modes, and Global Data Platform.
- **Deploying your workloads with Persistent volumes**
  This section shows you how you can provision Persistent volumes for application data persistence on IBM Fusion .

Storage deployment combinations from the user interface

Describes the supported storage deployment options available in the IBM Fusion user interface, including Fusion Data Foundation local, dynamic, and external modes, and Global Data Platform.

In the Services page of the IBM Fusion user interface, you can see only the services that can be deployed on the cluster. It is based on your current storage configuration. The different storage installation combinations are as follows:

- Fusion Data Foundation service on IBM Fusion
- Fusion Data Foundation service with Global Data Platform

Note: The Global Data Platform remote mount capability is available only for the following deployment modes:

- Standalone rack with Global Data Platform storage (newly installed with IBM Fusion 2.10.0 or later)
- Standalone rack with FDF (upgraded to IBM Fusion 2.10.0 or later / newly installed with IBM Fusion 2.10.0 or later)

Fusion Data Foundation service on IBM Fusion

Install Fusion Data Foundation in local, dynamic, or external modes. For the installation procedure, see Installing Fusion Data Foundation.

Fusion Data Foundation service with Global Data Platform

The deployment method for Global Data Platform depends on the installed version of Red Hat® OpenShift® Container Platform in your environment. The supported deployment methods are as follows:

Table 1. Global Data Platform deployment methods

Red Hat OpenShift Container Platform version	Deployment method
4.21 or later	1. Install Fusion Data Foundation version 4.21 or later in local, dynamic, or external modes. For instructions, see Installing Fusion Data Foundation. Important: The Fusion Data Foundation version 4.21 or later supports configuration with IBM Storage Scale. 2. After Fusion Data Foundation is installed successfully and the service is in a healthy state, go to Red Hat OpenShift Container Platform web console, to configure Fusion Data Foundation with IBM Storage Scale. For instructions, see Configuring with IBM Storage Scale. 3. In the Red Hat OpenShift Container Platform web console, go to Workloads > Deployments and verify the status of the pods in the `ibm-spectrum-scale` project. In a fresh installation, the deployment is expected to be scaled down to zero.
4.20 or earlier	1. From the Available section of the Services page, install Global Data Platform service. For instructions, see Installing Global Data Platform. 2. After Global Data Platform is installed successfully and the service is in a healthy state, go to Storage > Remote file systems and add remote file systems. For instructions, see Remote IBM Storage Scale storage cluster in IBM Fusion.

Deploying your workloads with Persistent volumes

This section shows you how you can provision Persistent volumes for application data persistence on IBM Fusion .

A PersistentVolume (PV) is a representation of storage volume in the IBM Fusion cluster that is provisioned by an administrator, or dynamically provisioned by Kubernetes, to fulfill a request made in a PersistentVolumeClaim (PVC).

PersistentVolume and PersistentVolumeClaim are independent of pod life cycles and preserve data through restarting, rescheduling, and even deleting Pods. For example, PersistentVolumes lets you store your WordPress platform data outside the containers. This way, even if the containers are deleted, their data persists.

With PersistentVolumeClaims, you can consume abstract storage resources. It is common that users need PersistentVolumes with varying properties, such as performance, for different problems. Cluster administrators need to be able to offer a variety of PersistentVolumes that differ in more ways than size and access modes, without exposing users to the details of how those volumes are implemented. For these needs, there is the *StorageClass* resource.

A PVC is a request for storage of a certain storage class by a user that can be fulfilled by a PV. There are two ways PVs may be provisioned: statically or dynamically.

This section explains how to deploy an application that uses PVs and PVCs to store data. A PersistentVolume (PV) is a piece of storage in the cluster that has been manually provisioned by an administrator, or dynamically provisioned by Kubernetes using a StorageClass. For example, WordPress with Persistent Volumes to access and work with the storage of your workloads.

Static
: A cluster administrator creates a number of PVs. They carry the details of the real storage, which is available for use by cluster users. For more information on creating Static provisioning, see Static provisioning of Persistent Volume in filesystem.

Dynamic
: Dynamic volume provisioning allows storage volumes to be created on-demand. In dynamic provisioning when a PersistentVolumeClaim is created, a PersistentVolume is dynamically provisioned based on the StorageClass configuration. For more information on creating dynamic provisioning, see Dynamic provisioning of Persistent Volume in filesystem.

Storage classes
: The storage class options for IBM Fusion by using Container Storage Interface driver:
For more information about IBM Storage Scale Container Native , see https://www.ibm.com/docs/en/scalecontainernative?topic=overview-introduction.

  For more information on how to create storage classes in IBM Spectrum® Storage Scale Erasure Code Edition (ECE) by using Container Network Interface (CNI), see IBM Storage Scale Container Storage Interface Driver storage class .

  The IBM Storage Scale Container Native Storage Access (CNSA) allows the deployment of the cluster file system in a Red Hat® OpenShift® cluster by using a remote mount attached file system.

- **Static provisioning of Persistent Volume in filesystem**
  Persistent Volume storage that is statically provisioned by an administrator.
- **Dynamic provisioning of Persistent Volume in filesystem**
  You can create and manage storage classes for dynamic provisioning of workloads.

Static provisioning of Persistent Volume in filesystem

Persistent Volume storage that is statically provisioned by an administrator.

1. Configure Persistent Volume manifest file with a `volumeHandle` as described in this example:

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: ibm-spectrum-fusion-pv-images
  labels:
    cns.isf.ibm.com/creator-fusion: ''
spec:
  capacity:
    storage: 10Gi
  csi:
    driver: spectrumscale.csi.ibm.com
```

```
    volumeHandle: 14544310586575975773;096E4651:61DFAFB3;path=/mnt/ibm-spectrum-scale-fs-mcgpfs3-
14544310586575975773/images
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  volumeMode: Filesystem
status:
  phase: Available
```

Here, the name is **ibm-spectrum-fusion-pv-images2**. The storage capacity is 10Gi. The **volumeHandle** value is **14544310586575975773;096E4651:61DFAFB3;path=/mnt/ibm-spectrum-scale-fs-mcgpfs3-14544310586575975773/images**.

2. Run the following apply command to create a Persistent Volume:

```
kubectl apply -f pv.yaml
```

Note:

- IBM Fusion creates static Persistent Volume templates for each filesystem.
- For IBM Fusion, static Persistent Volume for fileset supports backup but does not support backup for normal file or directory. If backup must be taken for normal file or directory, use dynamic Persistent Volume.

Dynamic provisioning of Persistent Volume in filesystem

You can create and manage storage classes for dynamic provisioning of workloads.

Storage classes in IBM Storage Scale

Note: IBM Fusion 2.7.1 or higher supports IBM Storage Scale 5.1.9. IBM Fusion 2.7 supports IBM Storage Scale 5.1.7.
There are two versions of storage classes that you can use with IBM Storage Scale. The default version works with all versions of IBM Storage Scale. For storage class examples, see Storage class (version 1) and Storage class (version 2). For the storage class to use with IBM Cloud Paks, IBM Storage Scale version (All), see Storage class to use with Cloud Paks.

Note: For IBM Storage Scale Container Storage Interface Driver volume expansion function, include **allowVolumeExpansion=true** in your Storage Classes. For information about how to set it in a storage class, see the example storage classes in this topic. For more information about volume expansion in CSI documentation, see Volume expansion.
The IBM Fusion detects the version of IBM Storage Scale, and creates storage class for each filesystem.

Procedure to create and apply a storage class YAML:

1. Create the following YAML and store it in the location of your choice. Retain this YAML until you create the storage class.
   Example of a storage class compatible with IBM Fusion deployed on Amazon Web Services ROSA:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: gp3-immediate
provisioner: ebs.csi.aws.com
parameters:
  encrypted: 'true'
  type: gp3
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

   Note: Ensure that you have a CSI compliant provisioner, such as **ebs.csi.aws.com**, and the volumeBindingMode is set to Immediate.
   Example of a storage class compatible with IBM Fusion deployed on Microsoft Azure ARO:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: managed-csi-immediate
provisioner: disk.csi.azure.com
parameters:
  skuname: Premium_LRS
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

   Note: Ensure that you have a CSI compliant provisioner, such as **disk.csi.azure.com**, and the volumeBindingMode is set to Immediate.
2. Run the following command to apply it:

```
oc apply -f <filename>
```

Storage class(version 1)
The default version that works with all versions of IBM Storage Scale. Example storage class (version 1) for IBM Storage Scale version (All):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-csi-fileset-dependent-new
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "ibm-spectrum-scale-fs-mcgpfs3-14544310586575975773"
  filesetType: "dependent"
  parentFileset: "primary-fileset-ibm-spectrum-scale-fs-mcgpfs3-14544310586575975773-2822920183822285462"
```

```
reclaimPolicy: Delete
allowVolumeExpansion: true
```

The file set path pattern is **primary-fileset-FS_CR_NAME-LocalClusterID**. If the fileset does not exist, then IBM Fusion creates it automatically from remote storage class.

For more information about the parameters, see Storage class.

Storage class(version 2)

This second storage class version requires IBM Storage Scale on Storage Cluster. Example storage class(version 2) for IBM Storage Scale version >= 5.1.3:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-csi-storageclass-version2
provisioner: spectrumscale.csi.ibm.com
parameters:
  version: "2"
  volBackendFs: "mygpfs"
reclaimPolicy: Delete
allowVolumeExpansion: true
```

The optional parameter list depends on the storage class type.

For more information about the additional parameters, see Volume expansion.

Storage classes in Fusion Data Foundation

There are two versions of storage classes that you can use with Fusion Data Foundation:

- Filesystem storage class **ibm-spectrum-fusion-mgmt-sc** and **ocs-storagecluster-cephfs**
- Block storage class **ocs-storagecluster-ceph-rbd**

Storage class to use with Cloud Paks

- Storage class to use with Cloud Paks for IBM Storage Scale
- Storage class to use with Cloud Paks for Fusion Data Foundation

IBM Storage Scale

The **ibm-storage-fusion-cp-sc** storage class is the default for use with IBM Cloud Paks, IBM Storage Scale (All). For IBM Cloud Paks, the permissions parameter is set to **shared: true** on the storage class.

The **ibm-storage-fusion-cp-sc** storage class is available for IBM Fusion:

```
NAME                          PROVISIONER                    RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   A
ibm-storage-fusion-cp-sc      spectrumscale.csi.ibm.com      Delete          Immediate              true                   2
```

```
oc get sc ibm-storage-fusion-cp-sc -o yaml
```

```
allowVolumeExpansion: true
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  creationTimestamp: "2023-08-06T13:07:55Z"
  name: ibm-storage-fusion-cp-sc
  resourceVersion: "4107984"
  uid: c8be0a2f-e379-4498-9051-ce8d4cde8d93
parameters:
  shared: "true"
  version: "2"
  volBackendFs: ibmspectrum-fs
provisioner: spectrumscale.csi.ibm.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

Data Foundation

Fusion Data Foundation defines storage classes for ReadWriteAfter (RWO) access and ReadWriteMany (RWX) access separately.
During the installation of IBM Cloud Paks, the **--block_storage_class** option must point to block storage class that supports ReadWriteAfter (RWO) access, which is **ocs-storagecluster-ceph-rbd**. The **--file_storage_class** option must point to a file storage class that supports ReadWriteMany (RWX) access, which is **ocs-storagecluster-cephfs**.

Note: For supported IBM Cloud Paks versions, see IBM Cloud Paks support for IBM Fusion.

Fusion services and capabilities

IBM Fusion Service provides scalable, integrated storage and data management solutions optimized for hybrid cloud and AI workloads on Red Hat® OpenShift®.

- **Fusion services**
  IBM Fusion Service provides scalable, integrated storage and data management solutions optimized for hybrid cloud and AI workloads on Red Hat OpenShift.
- **Disaster recovery**
  Disaster recovery offerings of IBM Fusion.

- **Multi-cluster IBM Fusion using Hosted Control Plane**
  You can create Hosted Control Plane clusters with OpenShift Virtualization on IBM Fusion and deploy IBM Fusion services on those clusters.
- **Red Hat OpenShift Virtualization**
  IBM Fusion enables the coexistence of containers and VMs, with OpenShift Virtualization allowing VMs to be managed as Kubernetes resources. Beyond VM hosting, Fusion and OpenShift Virtualization offer virtual machine mobility, backup and restore, and additional capabilities to ensure high availability and disaster recovery.
- **Advanced workloads**
  IBM Fusion is built to provide enterprise ready deployments for IBM Cloud Paks, IBM watsonx, IBM Maximo®, and Db2 Warehouse.

Fusion services

IBM Fusion Service provides scalable, integrated storage and data management solutions optimized for hybrid cloud and AI workloads on Red Hat® OpenShift®.

IBM Fusion supports preferred storage class selection during service installation by allowing each service to define an ordered list of storage classes in its FSD. The IBM Fusion UI reads this list, automatically selects the preferred storage class that exists in the cluster, and proceeds with installation without user input. If none of the preferred storage classes are available, the UI enables manual selection and records the user's choice in the service FSD.

The available services in IBM Fusion are as follows:

Global Data Platform

The Global Data Platform in IBM Fusion is a unified data management solution that simplifies access, sharing, and management of data across diverse environments. It offers scalable storage optimized for hybrid cloud and AI workloads, ensuring seamless data availability and mobility across multiple locations. It is a software-defined file and object storage that enables organizations to build a global data platform for artificial intelligence (AI), high-performance computing (HPC), advanced analytics, and other demanding workloads.

The Global Data Platform storage type provides the following features:

- File storage
- High availability through capacity-efficient erasure coding
- Metro and regional disaster recovery
- CSI snapshot support with built-in application consistency
- Encryption at rest
- Ability to mount file systems hosted by remote IBM Storage Scale clusters.

Important: IBM Fusion Access for Storage Area Network (SAN), Global Data Platform, and Content-Aware Storage are mutually exclusive services. If IBM Fusion Access for SAN is planned or installed, you cannot deploy the Global Data Platform or Content-Aware Storage services. Review your architectural requirements and deployment goals before installing these services.

Fusion Data Foundation

You can use Fusion Data Foundation as storage service in IBM Fusion. It enables you to manage Fusion Data Foundation services, and to deploy and scale storage clusters. The Fusion Data Foundation storage type provides the following features:

- Block, file, and object storage
- High availability through automatic data replication
- Metro and regional disaster recovery
- CSI snapshot support
- Encryption at rest

For more information about Fusion Data Foundation service, see Introduction to Fusion Data Foundation.
Important: In environments running Red Hat OpenShift Container Platform 4.21 or later and Fusion Data Foundation 4.21 or later, you can configure IBM Fusion Access for SAN in Fusion Data Foundation as an external system only if Global Data Platform and Content-Aware Storage services are not installed. These services are mutually exclusive services. Before planning or deploying IBM Fusion Access for SAN, ensure that your environment does not include Global Data Platform or Content-Aware Storage services.

Backup & Restore

The Backup & Restore service provides the following features:

- Policy-driven backup of applications that run on Red Hat OpenShift
- Change block detection for Ceph RBD block volumes. It greatly reduces backup times for applications using block mode volumes, including OpenShift Virtualization VMs based on RBD block volumes.
- Orchestration of application consistent online backups through recipes
- Multi-cluster Backup & Restore by using a hub and spoke topology

The Backup & Restore provides application-centric backup and data recovery. For more information about the service and its architecture, see Backup & Restore overview.

IBM Data Cataloging

The IBM Data Cataloging provides data insight for exabyte-scale heterogeneous file, object, backup, and archive storage on premises and in the cloud. The software connects to these data sources to rapidly ingest, consolidate, and index metadata for billions of files and objects. For more information about the IBM Data Cataloging service and its architecture, see Data Cataloging.

Content-Aware Storage (CAS)

It uses natural language processing (NLP) and AI technologies to extract semantic meaning from unstructured data, such as text, charts, graphs, and images. It allows for more efficient data retrieval and integration with AI workflows, particularly in retrieval-augmented generation (RAG) processes. For more information about the CAS service, see Content-Aware Storage (CAS).
Important: IBM Fusion Access for Storage Area Network (SAN), Global Data Platform, and Content-Aware Storage are mutually exclusive services. If IBM Fusion Access for SAN is planned or installed, you cannot deploy the Global Data Platform or Content-Aware Storage services. Review your architectural requirements and deployment goals before installing these services.

- **IBM Fusion services and platform support matrix**
  Support matrix of IBM Fusion services on the different deployment platforms of IBM FusionSupport matrix of IBM Fusion services on the Bare Metal deployment platform of IBM Fusion HCI.
- **Fusion Data Foundation**
  Welcome to the official documentation for IBM Fusion Data Foundation 4.21.
- **Global Data Platform**
  Learn how to install and configure Global Data Platform service.
- **Backup & Restore**
  The Backup & Restore service protects your workloads with tailored backups by using a hub and spoke model. You can restore your data rapidly from local snapshots without having to rely on external storage. Backups can be transferred to external object storage solutions like IBM Cloud, Microsoft Azure, Amazon Web Services, and any S3-compliant object storage. It allows the protection of the IBM Fusion Backup & Restore service itself to a S3 bucket.
- **Data Cataloging**
  Companies need the ability to use unstructured data to meet their business priorities.
- **Content-Aware Storage (CAS)**
  The IBM Fusion Content-Aware Storage service is engineered to enhance the value of customers' AI applications, such as retrieval augmented generation (RAG) for faster time to insights, reduced cost, improved performance, enhanced security, and streamlined operations.
- **Lakehouse-Table [Technology preview]**
  The Lakehouse-Table service is a Kubernetes-native solution for building and managing lakehouse infrastructure on IBM Fusion. It provides a simplified experience for administrators to provision and manage Data Lakehouse capabilities.
- **Deploying IBM Fusion services using CLI**
  Use the instructions to deploy the IBM Fusion services using the command-line interface (CLI).

IBM Fusion services and platform support matrix

Support matrix of IBM Fusion services on the different deployment platforms of IBM FusionSupport matrix of IBM Fusion services on the Bare Metal deployment platform of IBM Fusion HCI.

IBM Fusion

Table having a matrix of what IBM Fusion service is supported on which deployment platforms of IBM Fusion

Amazon Web Services

IBM Fusion Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	IBM Data Cataloging	Content-Aware Storage (CAS)
Self-managed OpenShift® Container Platform on Amazon Web Services	Yes	Yes	No	Yes	Yes	No
Amazon Web Services ROSA	Yes	No	Yes	Yes	Yes	Yes

Microsoft Azure

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	IBM Data Cataloging	Content-Aware Storage (CAS)
Self-managed OpenShift Container Platform on Microsoft Azure	Yes	Yes	No	Yes	Yes	No
Microsoft Azure ARO	Yes	Yes	No	Yes	Yes	No

IBM Cloud®

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	IBM Data Cataloging	Content-Aware Storage (CAS)
ROKS IBM Classic	No	Yes	No	Yes	Yes	No
ROKS VPC	No	Yes	No	Yes	Yes	No

IBM Power Systems

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	IBM Data Cataloging	Content-Aware Storage (CAS)
IBM Power Systems	Yes	Yes	Yes	Yes	Yes	No

Bare Metal

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	IBM Data Cataloging	Content-Aware Storage (CAS)
Bare Metal	Yes	Yes	Yes	Yes	Yes	Yes

Linux on IBM Z

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	IBM Data Cataloging	Content-Aware Storage (CAS)
Linux on IBM Z	Yes	Yes	Yes	Yes	Yes	No

zCX

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	IBM Data Cataloging	Content-Aware Storage (CAS)
zCX	Yes	Yes	Yes	Yes	Yes	No

On-premises VMware

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	IBM Data Cataloging	Content-Aware Storage (CAS)
On-premises VMware	Yes	Yes	Yes	Yes	Yes	Yes

Google Cloud

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	IBM Data Cataloging	Content-Aware Storage (CAS)
Google Cloud	Yes	Yes	No	Yes	No	No

Fusion Data Foundation

Welcome to the official documentation for IBM Fusion Data Foundation 4.21.

This documentation provides information on Fusion Data Foundation, including:

- Planning considerations required before deployment
- Step-by-step installation and deployment guidance
- Detailed configuration instructions for new features and enhancements
- Recommendations and procedures for managing the deployment
- Guidance on monitoring cluster health, metrics, and alerts
- Upgrade and migration procedures from previous versions
- Troubleshooting tips and best practices

The Fusion Data Foundation 4.21 release introduces a set of new features, enhancements, and bug fixes designed to improve performance, usability, and integration capabilities across supported environments.

Remember: Each IBM Fusion release documentation includes content only for the latest supported version of Fusion Data Foundation. To access documentation for earlier Fusion Data Foundation versions, refer to the corresponding IBM Fusion release documentation. For version compatibility details, see the Support matrix. It is recommended to use the latest Fusion Data Foundation version to ensure full support and access to the latest features and fixes.

- **Release notes**
  IBM Fusion Data Foundation 4.21 is now available. New features, enhancements, and known issues that pertain to Fusion Data Foundation 4.21 are included in this section.
- **Introduction to Fusion Data Foundation**
  IBM Fusion Data Foundation is a highly integrated collection of cloud storage and data services for Red Hat OpenShift Container Platform. It is available as a IBM Fusion storage service to facilitate simple deployment and management.
- **Fusion Data Foundation architecture**
  Use this information to understand the Fusion Data Foundation architecture.
- **Planning Fusion Data Foundation deployment**
  Use this information for important considerations when planning your IBM Fusion Data Foundation deployment.
- **Installing Fusion Data Foundation**
  Install Fusion Data Foundation in local, dynamic, external, client, or provider modes.
- **Uninstalling Fusion Data Foundation**
  Use the following steps to uninstall Data Foundation by using command line interface.
- **Configuring Fusion Data Foundation**
  Configure Fusion Data Foundation to set up storage services that support scalable, resilient, and flexible data management for OpenShift applications.
- **Preparing to connect to an external KMS server in Fusion Data Foundation**
  Procedure to prepare for the connection to an external KMS from Fusion Data Foundation.
- **Deploying Data Foundation in external mode**
  Fusion Data Foundation can make services from an external IBM Storage Ceph cluster available for consumption through OpenShift Container Platform clusters.
- **Managing and allocating resources**
  Understand how to create, configure, and allocate storage to core services or hosted applications in IBM Fusion Data Foundation.
- **Managing hybrid and multicloud resource**
  Use this information to learn how to manage hybrid and multicloud resources.
- **Replacing nodes**
  Explains how to safely replace a node in a Fusion Data Foundation cluster.
- **Replacing devices**
  Explains how to safely replace storage devices for Fusion Data Foundation.
- **Monitoring Fusion Data Foundation**
  Learn how to monitor IBM Fusion Data Foundation using the Block and File, and Object dashboards.
- **Fusion Data Foundation service install and upgrade issues**
  Use this troubleshooting information to resolve install and upgrade problems that are related to Fusion Data Foundation service.
- **Troubleshooting**
  Administrators can use this troubleshooting information to understand how to troubleshoot and fix their IBM Fusion Data Foundation cluster.
- **Fusion Data Foundation service error scenarios**
  Use these troubleshooting information to know the problem and workaround when install or configure Fusion Data Foundation service.

Release notes

IBM Fusion Data Foundation 4.21 is now available. New features, enhancements, and known issues that pertain to Fusion Data Foundation 4.21 are included in this section.

- **New features**
  Each release offers new functionalities and improvements. Read this information for a high-level summary of the new features introduced in the Fusion Data Foundation 4.21 release.
- **Enhancements**
  Each release offers new functionalities and improvements. Read this information for a high-level summary of major enhancements introduced in the Fusion Data Foundation 4.21 release.
- **Known issues**
  This section describes the known issues in Fusion Data Foundation 4.21.

New features

Each release offers new functionalities and improvements. Read this information for a high-level summary of the new features introduced in the Fusion Data Foundation 4.21 release.

- CNSA integration with Fusion Data Foundation
- IBM Fusion Access for Storage Area Network (SAN) integration with Fusion Data Foundation
- Disaster Recovery
- Networking

CNSA integration with Fusion Data Foundation

IBM Fusion 2.13 introduces integration between Fusion Data Foundation and Container Native Storage Access (CNSA) by using the remote mount feature. This feature enables Fusion Data Foundation to deploy and manage the required IBM Storage Scale or Global Data Platform resources during installation. It also allows administrators to configure and attach external IBM Storage Scale clusters directly from the Red Hat® OpenShift® Container Platform 4.21 user interface.

The integration simplifies the installation, upgrade, and lifecycle management of remote mount functionality across IBM Fusion deployments, supports seamless upgrades and migrations from earlier releases, and ensures continuity for existing workloads while enabling a unified, UI-driven storage management experience.

IBM Fusion Access for Storage Area Network (SAN) integration with Fusion Data Foundation

IBM Fusion 2.13 introduces IBM Fusion Access for SAN as a Fusion Data Foundation capability. This feature integrates the existing SAN storage operator into the Fusion Data Foundation workflow, providing a unified external SAN integration for OpenShift Virtualization on Red Hat OpenShift Container Platform 4.21.

This integration simplifies deployment and management of external SAN storage by enabling it to be configured and managed through Fusion Data Foundation.

Disaster Recovery

Hybrid cloud support for secondary Fusion Data Foundation clusters
: Support has been added for pairing an on-premises Fusion Data Foundation deployment with a secondary cluster in the cloud, enabling hybrid disaster recovery and workload flexibility. This capability allows on-prem environments such as Bare Metal and vSphere to integrate with cloud platforms including Azure and Azure Red Hat OpenShift (ARO), expanding Fusion Data Foundation's support for hybrid deployment scenarios.

  For more information, see Hybrid cloud Regional-DR solution.

Multi-volume consistency for disaster recovery
: Multi-volume consistency group support is now fully supported again for both, RBD and CephFS volumes. Consistency groups are used by default for all newly protected applications. Already protected non-CG applications on an upgraded cluster applications should be unprotected and protected again to migrate to consistency groups.

Support for CephFS volumes using non-default StorageClasses
: Fusion Data Foundation DR solution supports data replication for CephFS volumes that use multiple or non-default StorageClasses, such as replica-2 file storage. This enhancement enables disaster recovery for workloads backed by customized CephFS configurations, including cases where different pools or secrets require separate StorageClass, VolumeSnapshotClass, and VolumeReplicationClass resources.

Enhanced disaster recovery capabilities in the RHACM KubeVirt UI
: **Disaster recovery integration for virtual machines**

  The Red Hat Advanced Cluster Management for Kubernetes (RHACM) KubeVirt UI now integrates Fusion Data Foundation disaster recovery, allowing users to view VM protection status and initiate DR protection directly from the VM view. This capability is available for both GitOps-managed and discovered VMs, with disabled actions providing guidance when prerequisites are not met.

  Improved visibility for failover and relocate operations

  The RHACM UI now surfaces detailed, step-by-step progression for disaster recovery failover and relocate workflows using progression data from Ramen. Enhanced modals show each operation phase, status indicators, and any associated errors, improving transparency, troubleshooting speed, and overall observability of DR processes.

Networking

IPv6 support for Multus networks
: Multus supports IPv6, allowing customers to configure their Multus networks as IPv4-only or IPv6-only. This enhancement improves network flexibility and supports modern dual-stack deployment needs. Both Multus networks must use the same IP family when multiple networks are configured.

IPv6 support for external mode

Fusion Data Foundation clusters deployed in external mode now support single-stack IPv6, in addition to IPv4, enabling deployments in environments where IPv6 is required with limited IPv4 availability. This enhancement aligns Fusion Data Foundation external mode with OpenShift's IPv6 capabilities and expands deployment flexibility for customers operating in IPv6-centric infrastructures.

Enhancements

Each release offers new functionalities and improvements. Read this information for a high-level summary of major enhancements introduced in the Fusion Data Foundation 4.21 release.

- HCP clusters
- Disaster Recovery
- Multus support for existing Fusion Data Foundation clusters
- Multicloud Object Gateway
- Performance Plus enabled for Azure deployments
- Monitoring enhancements
- Enhanced vCPU requirements for IBM Z and LinuxONE
- Taint Handling in Fusion Data Foundation

HCP clusters

Fusion Data Foundation support for HCP clusters
: Fusion Data Foundation is supported on hosted (HCP) clusters running OpenShift® Container Platform versions N, N-1, N-2, N-3, and N-4, where N represents the OpenShift Container Platform version deployed on the hub cluster.

Fusion Data Foundation HCP clusters deployment support for Amazon Web Services (AWS) and IBM Z
: Fusion Data Foundation services on OpenShift HCP clusters are now supported on Amazon Web Services (AWS) and IBM Z, extending existing support beyond Bare Metal deployments.

Disaster Recovery

Enhanced protected applications list view for disaster recovery
: The protected Applications page shows managed (ApplicationSet and Subscription) and discovered applications together, providing a complete view of all workloads protected by disaster recovery. Multi-selection is enabled in the list view, laying the groundwork for grouped failover, relocate, and disable-DR operations for managed applications.

Automatic resource cleanup for discovered VMs during disaster recovery operations
: Disaster recovery workflows now automatically remove resources associated with DR-protected virtual machines during failover or relocate operations. This automation applies only to the discovered virtual machines enrolled for protection through Red Hat® Advanced Cluster Management for Kubernetes Fleet Virtualization and helps reduce manual steps while improving transition efficiency.

  For more information, see Virtual machine garbage collection automation.

Multus support for existing Fusion Data Foundation clusters

Fusion Data Foundation now supports configuring Multus on existing clusters, enabling customers to introduce public and private storage networks after initial deployment. This enhancement allows administrators to add or expand Network Attachment Definitions (NADs), such as separating client-facing and replication traffic even when a cluster is already using a single Multus network. This provides greater flexibility for network isolation, performance tuning, and aligning storage traffic with organizational requirements.

For more information, see Enabling Multus support on an existing Fusion Data Foundation cluster.

Multicloud Object Gateway

Minimal IAM support for Multicloud Object Gateway
: Minimal IAM support is now available in the Multicloud Object Gateway to provide improved access control for multi-tenant environments. This enhancement enables administrators to define and manage user-level permissions more effectively, supporting scenarios where data scientists and other user groups require isolated and governed access to object storage resources. The feature lays the groundwork for more secure, flexible multi-tenant deployments by ensuring that users interact only with the buckets and operations they are authorized to access.

  For more information, see Creating and managing IAM user.

Enhanced Multicloud Object Gateway database backup
: Multicloud Object Gateway database backup is enhanced to provide more reliable protection against data loss and to complement the high-availability capabilities. The update adds support for automated scheduled backups with configurable frequency (daily, weekly, or monthly) and retention settings along with on-demand backup options through the MCG CLI or CRD. The UI now displays the time of the most recent backup and offers advanced configuration for Multicloud Object Gateway backup management. The system supports internal, external, and standalone Multicloud Object Gateway deployments by automatically detecting or allowing selection of snapshot StorageClasses.

  To configure Multicloud Object Gateway database backup, refer to the deployment procedures for your specific platform.

Developer access to the Fusion Data Foundation object browser
: Developer-level users can now be granted access to the Fusion Data Foundation object browser, enabling them to view and manage only their own object storage resources with restricted permissions. This enhancement supports flexible, role-appropriate access control and allows development teams to work more independently while preserving administrative boundaries.

  For more information, see Accessing the object browser.

Improved Multicloud Object Gateway performance using the secondary database instance
: Multicloud Object Gateway performance has been improved by directing read-only database operations to the existing secondary database instance, reducing load on the primary and improving overall responsiveness. By separating read and write activity, the system minimizes the impact of heavy read queries, such as lifecycle and background worker operations on write performance and object I/O. This enhancement leverages the high-availability secondary instance already deployed in recent versions to deliver more consistent and efficient database behavior.

  For more information, see the DB pods section within Responsibilities and resources.

Support for internal mode RGW with object browser
: The object browser now supports browsing, uploading, downloading, and managing RGW object. Operations include listing and creating buckets, navigating objects in a folder-like view, uploading files or directories, downloading objects, generating presigned URLs, viewing metadata, filtering and sorting results, previewing content, and creating prefixes. Bucket-level capabilities such as configuring expiration and access policies are also available.

  For more information, see Creating and managing buckets using the object browser.

Performance Plus enabled for Azure deployments

New Azure-based deployments now automatically enable Performance Plus for eligible Standard SSD and premium SSD disks (≥513 GiB) when using the Azure Disk CSI driver. This provides higher IOPS and throughput for Fusion Data Foundation clusters on Azure and helps to achieve the best possible storage performance.

For more information about how to use, see the Prerequisites section in Configuring Fusion Data Foundation in dynamic mode.

Monitoring enhancements

Additional NooBaa bucket and replication metrics
: Introduces **`bucket_last_cycle_total_objects_num`**, **`bucket_last_cycle_replicated_objects_num`**, and **`bucket_last_cycle_error_objects_num`**, along with expanded bucket-usage gauges for object and size quotas.

Clearer NooBaa capacity alerting
: Adds concise runbooks for the **`NooBaaSystemCapacityWarning85`**, **`95`**, and **`100`** alerts, outlining how capacity is calculated and the required mitigation steps.

  For more information, see NooBaaSystemCapacityWarning85, NooBaaSystemCapacityWarning95, and NooBaaSystemCapacityWarning100.

New MDS xattr-latency alert
: Adds the **`CephXattrSetLatency`** alert for elevated **`setxattr`** latency, with guidance for evaluating and adjusting MDS resource allocation.

  For more information, see CephXattrSetLatency.

Enhanced vCPU requirements for IBM Z and LinuxONE

Fusion Data Foundation now automatically detects IBM Z and LinuxONE architectures and applies optimized vCPU requirements specific to these platforms. Deployments on IBM Z and LinuxONE no longer require the same number of vCPUs as other architectures, reducing resource overhead and improving deployment efficiency.

Taint Handling in Fusion Data Foundation

Customers using taints and toleration need to update their configuration using the guidance in the knowledgebase article How to add toleration for the "non-ocs" taints to the OpenShift Data Foundation pods.

Known issues

This section describes the known issues in Fusion Data Foundation 4.21.

- Disaster recovery
- Multicloud Object Gateway
- Ceph
- CSI driver
- IBM Storage Scale operator
- IBM Fusion Access for SAN
- Fusion Data Foundation console

Disaster recovery

CIDR range does not persist in **`csiaddonsnode`** object when the respective node is down
: When a node is down, the Classless Inter-Domain Routing (CIDR) information disappears from the **`csiaddonsnode`** object. This impacts the fencing mechanism when it is required to fence the impacted nodes.

  Workaround:

  Collect the CIDR information immediately after the **`NetworkFenceClass`** object is created.

  (DFBUGS-2948)

DRPCs protect all persistent volume claims created on the same namespace
: The namespaces that host multiple disaster recovery (DR) protected workloads protect all the persistent volume claims (PVCs) within the namespace for each DRPlacementControl resource in the same namespace on the hub cluster that does not specify and isolate PVCs based on the workload using its **`spec.pvcSelector`** field.

This results in PVCs that match the DRPlacementControl **`spec.pvcSelector`** across multiple workloads. Or, if the selector is missing across all workloads, replication management to potentially manage each PVC multiple times and cause data corruption or invalid operations based on individual DRPlacementControl actions.

Workaround:

Label PVCs that belong to a workload uniquely, and use the selected label as the DRPlacementControl **`spec.pvcSelector`** to disambiguate which DRPlacementControl protects and manages which subset of PVCs within a namespace. It is not possible to specify the **`spec.pvcSelector`** field for the DRPlacementControl using the user interface, hence the DRPlacementControl for such applications must be deleted and created using the command line.

Result:

PVCs are no longer managed by multiple DRPlacementControl resources and do not cause any operation and data inconsistencies.

(DFBUGS-1749)

Disabled **`PeerReady`** flag prevents changing the action to failover

The DR controller executes full reconciliation as and when needed. When a cluster becomes inaccessible, the DR controller performs a sanity check. If the workload is already relocated, this sanity check causes the **`PeerReady`** flag associated with the workload to be disabled, and the sanity check does not complete due to the cluster being offline. As a result, the disabled **`PeerReady`** flag prevents you from changing the action to failover.

Workaround:

Use the command-line interface to change the DR action to failover despite the disabled **`PeerReady`** flag.

(DFBUGS-665)

Information about **`lastGroupSyncTime`** is lost after hub recovery for the workloads which are primary on the unavailable managed cluster

Applications that are previously failed over to a managed cluster do not report a **`lastGroupSyncTime`**, thereby causing the trigger of the alert **`VolumeSynchronizationDelay`**. This is because when the ACM hub and a managed cluster that are part of the DRPolicy are unavailable, a new ACM hub cluster is reconstructed from the backup.

Workaround:

If the managed cluster to which the workload was failed over is unavailable, you can still failover to a surviving managed cluster.

(DFBUGS-376)

MCO operator reconciles the **`veleroNamespaceSecretKeyRef`** and **`CACertificates`** fields

When the Fusion Data Foundation operator is upgraded, the **`CACertificates`** and **`veleroNamespaceSecretKeyRef`** fields under **`s3StoreProfiles`** in the Ramen config are lost.

Workaround:

If the Ramen config has the custom values for the **`CACertificates`** and **`veleroNamespaceSecretKeyRef`** fields, then set those custom values after the upgrade is performed.

(DFBUGS-440)

For discovered apps with CephFS, sync stop after failover

For CephFS-based workloads, synchronization of discovered applications may stop at some point after a failover or relocation. This can occur with a **`Permission Denied`** error reported in the **`ReplicationSource`** status.

Workaround:

- **For Non-Discovered Applications**
  1. Delete the VolumeSnapshot.

     ```
     oc delete volumesnapshot -n <vrg-namespace> <volumesnapshot-name>
     ```

     The snapshot name usually starts with the PVC name followed by a timestamp.

  2. Delete the VolSync Job.

     ```
     oc delete job -n <vrg-namespace> <pvc-name>
     ```

     The job name matches the PVC name.

- **For Discovered Applications**

  Use the same steps, except **`<namespace>`** refers to the **application workload namespace**, not the VRG namespace.

- **For Workloads using Consistency Groups**

  - Delete the ReplicationGroupSource:

    ```
    oc delete replicationgroupsource -n
    <namespace> <name>
    ```

  - Delete all VolSync jobs in that namespace:

    ```
    oc delete jobs --all -n
    <namespace>
    ```

    In this case, **`<namespace>`** refers to the namespace of the workload (either discovered or not), and **`<name>`** refers to the name of the ReplicationGroupSource resource.

(DFBUGS-2883)

Remove DR option is not available for discovered apps on the Virtual machines page

The Remove DR option is not available for discovered applications listed on the Virtual machines page.

Workaround:

1. Add the missing label to the DRPlacementControl.

```
{{oc label drplacementcontrol <drpcname> \
odf.console.selector/resourcetype=virtualmachine \
-n openshift-dr-ops}}
```

2. Add the `PROTECTED_VMS` recipe parameter with the virtual machine name as its value.

```
{{oc patch drplacementcontrol <drpcname> \
-n openshift-dr-ops \
--type='merge' \
-p '{"spec":{"kubeObjectProtection":{"recipeParameters":{"PROTECTED_VMS":["<vm-name>"]}}}}'}}
```

(DFBUGS-2823)

DR Status is not displayed for discovered apps on the Virtual machines page
DR Status is not displayed for discovered applications listed on the Virtual machines page.

Workaround:

1. Add the missing label to the DRPlacementControl.

```
{{oc label drplacementcontrol <drpcname> \
odf.console.selector/resourcetype=virtualmachine \
-n openshift-dr-ops}}
```

2. Add the `PROTECTED_VMS` recipe parameter with the virtual machine name as its value.

```
{{oc patch drplacementcontrol <drpcname> \
-n openshift-dr-ops \
--type='merge' \
-p '{"spec":{"kubeObjectProtection":{"recipeParameters":{"PROTECTED_VMS":["<vm-name>"]}}}}'}}
```

(DFBUGS-2822)

Secondary PVCs aren't removed when DR protection is removed for discovered apps
On the secondary cluster, CephFS PVCs linked to a workload are usually managed by the VolumeReplicationGroup (VRG). However, when a workload is discovered using the Discovered Applications feature, the associated CephFS PVCs are not marked as VRG-owned. As a result, when the workload is disabled, these PVCs are not automatically cleaned up and become orphaned.

Workaround:

To clean up the orphaned CephFS PVCs after disabling DR protection for a discovered workload, manually delete them using the following command:

```
oc delete pvc <pvc-name> -n <pvc-namespace>
```

(DFBUGS-2827)

Multicloud Object Gateway

Unable to create new OBCs using Multicloud Object Gateway
When provisioning an NSFS bucket by using ObjectBucketClaim (OBC), the default filesystem path is expected to use the bucket name. However, if path is set in `OBC.Spec.AdditionalConfig`, it should take precedence. This behavior is currently inconsistent, resulting in failures when creating new OBCs.

(DFBUGS-3817)

Ceph

Poor CephFS performance on stretch clusters
Workloads with many small metadata operations might exhibit poor performance because of the arbitrary placement of metadata server pods (MDS) on multi-site Fusion Data Foundation clusters.

(DFBUGS-1753)

OSD pods restart during add capacity
OSD pods restart after performing cluster expansion by adding capacity to the cluster. However, no impact to the cluster is observed apart from pod restarting.

(DFBUGS-1426)

Ceph becomes inaccessible and IO is paused when connection is lost between the two data centers in stretch cluster
When two data centers lose connection with each other but are still connected to the Arbiter node, there is a flaw in the election logic that causes an infinite election among Ceph Monitors. As a result, the Monitors are unable to elect a leader and the Ceph cluster becomes unavailable. Also, IO is paused during the connection loss.

Workaround:

Shutdown the monitors of any one data zone by bringing down the zone nodes. Additionally, you can reset the connection scores of surviving Monitor pods.

As a result, Monitors can form a quorum and Ceph becomes available again and IOs resumes.

(DFBUGS-425)

SELinux relabelling issue with a very high number of files
When attaching volumes to pods in Red Hat® OpenShift® Container Platform, the pods sometimes do not start or take an excessive amount of time to start. This behavior is generic and it is tied to how SELinux relabelling is handled by Kubelet. This issue is observed with any filesystem based volumes having very high file counts. In Fusion Data Foundation, the issue is seen when using CephFS based volumes with a very high number of files. There are multiple ways to work around

this issue. Depending on your business needs you can choose one of the workarounds from the knowledgebase solution https://access.redhat.com/solutions/6221251.

(RFE-3327)

CSI driver

Sync stops after PVC deselection

When a PersistentVolumeClaim (PVC) is added to or removed from a group by modifying its label to match or unmatch the group criteria, sync operations may unexpectedly stop. This occurs due to stale protected PVC entries remaining in the VolumeReplicationGroup (VRG) status.

Workaround:

Manually edit the VRG's status field to remove the stale protected PVC:

```
oc edit vrg <vrg-name> -n <vrg-namespace> --subresource=status
```

(DFBUGS-4012)

IBM Storage Scale operator

Scale Dashboard reports "Healthy" connection despite total storage I/O failure on worker nodes

The IBM Storage Scale (CNSA) operator fails to report a connection failure when the storage data path is severed between OpenShift worker nodes and the remote Scale cluster. While the data plane is functionally dead, the RemoteCluster custom resource (CR) and the Scale Dashboard UI continue to report a status of "Healthy/Available." This occurs because the health check relies solely on the reachability of the remote Scale REST API from the operator pod, ignoring the actual mount state and heartbeat status of the compute (daemon) pods on the worker nodes.

Workaround:

Perform manual verification by running **`mmlscluster`** from the CLI or run an **`ls`** command directly on the mount point to confirm the actual storage health and path status.

(DFBUGS-6498)

IBM Fusion Access for SAN

PersistentVolumeClaims page shows inaccurate used capacity for copy-on-write cloned volumes

In the OpenShift web console, the PersistentVolumeClaims page includes a Used column that displays the used capacity of a volume. For volumes created using a StorageClass with copy-on-write cloning enabled, the Used value may be inaccurate and should be ignored. This behavior is only applicable to PersistentVolumeClaims created from a snapshot.

There is currently no workaround for this issue.

(DFBUGS-7420)

LUN group shows unhealthy status during creation

The LUN group status does not transition to a healthy state because the OpenShift web console cannot accurately determine its health. The file system resource takes approximately 30 minutes to reach a healthy state.

There is currently no workaround for this issue.

(DFBUGS-7390)

Fusion Data Foundation console

UI shows **`WaitOnUserCleanUp`** even when automatic cleanup is enabled

The UI incorrectly displays the **`WaitOnUserCleanUp`** status even when automatic cleanup is enabled for VMs. This occurs because the UI relies only on the **`phase`** and **`progression`** fields of the **`DRPlacementControl`** to determine cleanup behavior and does not evaluate the more granular **`AutoCleanup`** condition that explicitly indicates automatic cleanup.

Workaround:

There is no manual workaround required. This state is transient and clears automatically once the **`progression`** field advances to **`Completed`**. Manual cleanup should be avoided unless the **`AutoCleanup`** condition and its corresponding **`reason`** in the **`DRPlacementControl`** or VRG status indicate otherwise.

During automatic cleanup, the UI may briefly present a misleading status, which can cause temporary confusion until the cleanup completes.

(DFBUGS-5824)

DRPlacementControl shows **`ProtectionError`** even after successful relocation

When a relocation completes, the **`DRPlacementControl`** may continue to display a **`ProtectionError`** status. This occurs because the **`Protected`** condition in the DRPlacementControl status incorrectly reports an **`Error`** state, even though the relocation has finished (**`phase: Relocated`**, **`progression: Completed`**).

Workaround:

No direct workaround is available. Wait until retrying the **`NoClusterDataConflict`** condition is met.

The DR status in the UI remains in the **`ProtectionError`** state until the data conflict is resolved.

(DFBUGS-5823)

UI shows "Unauthorized" error and blank screen with loading temporarily during Fusion Data Foundation operator installation

During Fusion Data Foundation operator installation, sometimes the `InstallPlan` transiently goes missing which causes the page to show unknown status. This does not happen regularly. As a result, the messages and title go missing for a few seconds.

(DFBUGS-3574)

Introduction to Fusion Data Foundation

IBM Fusion Data Foundation is a highly integrated collection of cloud storage and data services for Red Hat OpenShift Container Platform. It is available as a IBM Fusion storage service to facilitate simple deployment and management.

Fusion Data Foundation services are primarily made available to applications in the form of storage classes that represent the following components:

- Block storage devices, catering primarily to database workloads, for example, Red Hat OpenShift Container Platform logging and monitoring, and PostgreSQL. Important: Block storage should be used for any workload only when it does not require sharing the data across multiple containers.
- Shared and distributed file system, catering primarily to software development, messaging, and data aggregation workloads, for example, Jenkins build sources and artifacts, Wordpress uploaded content, Red Hat OpenShift Container Platform registry, and messaging using JBoss AMQ.
- Multicloud object storage, featuring a lightweight S3 API endpoint that can abstract the storage and retrieval of data from multiple cloud object stores.
- On-premises object storage, featuring a robust S3 API endpoint that scales to tens of petabytes and billions of objects, primarily targeting data intensive applications, for example, the storage and access of row, columnar, and semi-structured data with applications like Spark, Presto, Red Hat AMQ Streams (Kafka), and even machine learning frameworks like TensorFlow and Pytorch.

Note:
Running PostgresSQL workload on CephFS persistent volume is not supported and it is recommended to use RADOS Block Device (RBD) volume.

Fusion Data Foundation version 4.x integrates a collection of software projects, including:

- Ceph, providing block storage, a shared and distributed file system, and on-premises object storage.
- Ceph CSI, to manage provisioning and lifecycle of persistent volumes and claims.
- NooBaa, providing a Multicloud Object Gateway (MCG).
- Fusion Data Foundation, rook-ceph, and NooBaa operators to initialize and manage Fusion Data Foundation services.

Fusion Data Foundation architecture

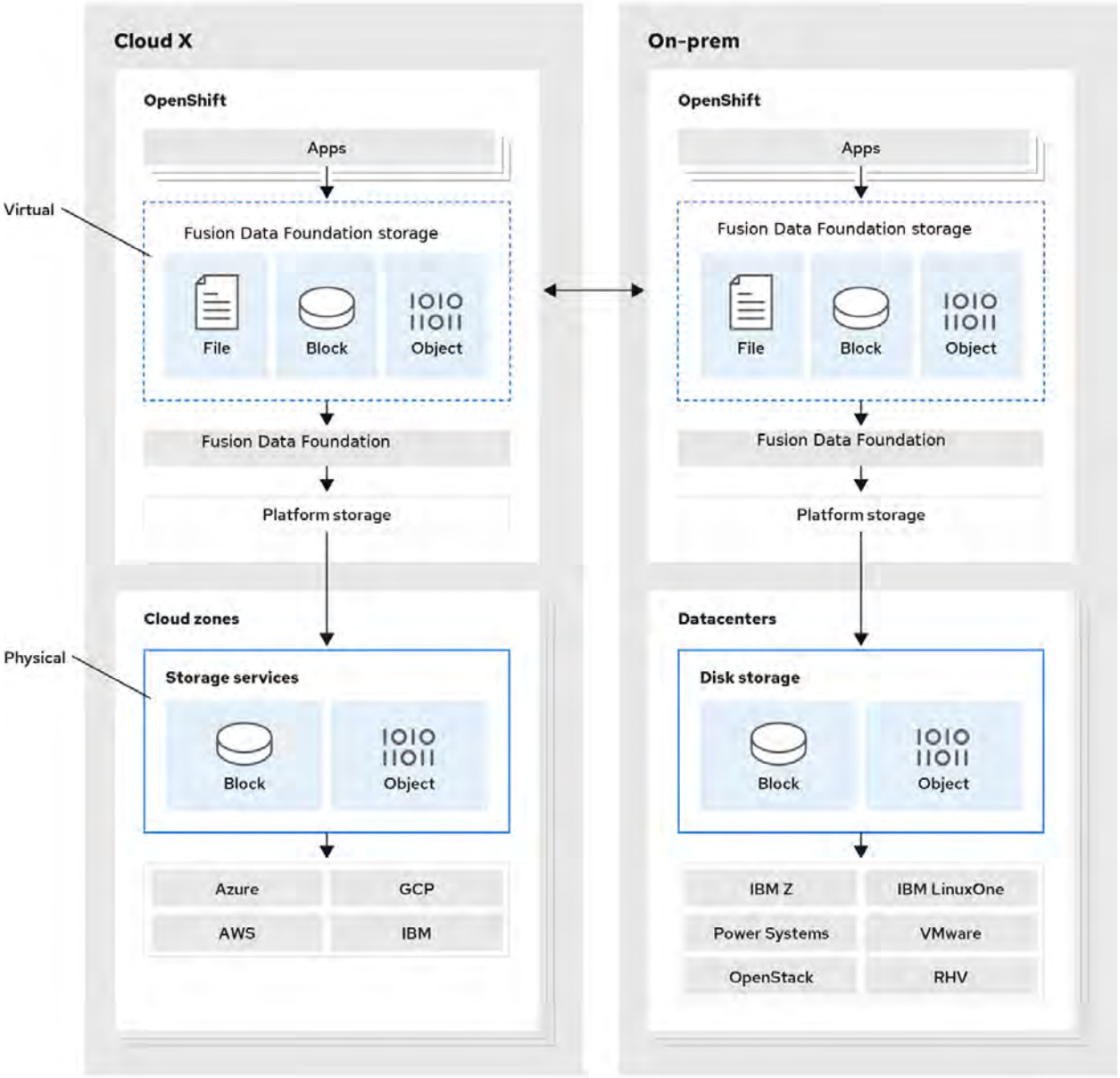
Use this information to understand the Fusion Data Foundation architecture.

- **An overview of Fusion Data Foundation architecture**
  IBM Fusion Data Foundation provides services for, and can run internally from Red Hat OpenShift Container Platform.
- **Fusion Data Foundation Operators**
  IBM Fusion Data Foundation is comprised of three Operator Lifecycle Manager (OLM) operator bundles, deploying four operators which codify administrative tasks and custom resources so that task and resource characteristics can be automated.
- **Fusion Data Foundation installation overview**
  Fusion Data Foundation consists of multiple components managed by multiple operators.
- **Fusion Data Foundation upgrade overview**
  As an operator bundle managed by the Operator Lifecycle Manager (OLM), Fusion Data Foundation leverages its operators to perform high-level tasks of installing and upgrading the product through `ClusterServiceVersion` (CSV) custom resources (CRs).
- **Multus architecture for Fusion Data Foundation**
  Learn about supported processes and configurations for using the OpenShift multi-network plugin (Multus) with Fusion Data Foundation, including requirements, processes, and features.

An overview of Fusion Data Foundation architecture

IBM Fusion Data Foundation provides services for, and can run internally from Red Hat OpenShift Container Platform.

Figure 1. Fusion Data Foundation architecture



Fusion Data Foundation supports deployment into Red Hat OpenShift Container Platform clusters deployed on Installer Provisioned Infrastructure or User Provisioned Infrastructure. For details about these two approaches, see Discover > Architecture in Red Hat OpenShift Container Platform product documentation. For more information about interoperability of components for the Fusion Data Foundation and Red Hat OpenShift Container Platform, see the interoperability matrix Knowledgebase article on the Red Hat Customer Portal.

For information about the architecture and lifecycle of OpenShift Container Platform, see Discover > Architecture in Red Hat OpenShift Container Platform product documentation.

Fusion Data Foundation Operators

IBM Fusion Data Foundation is comprised of three Operator Lifecycle Manager (OLM) operator bundles, deploying four operators which codify administrative tasks and custom resources so that task and resource characteristics can be automated.

The three OLM operator bundles are:

- Fusion Data Foundation
  - `odf-operator`
- Container Storage
  - `ocs-operator`
  - `rook-ceph-operator`
- Multicloud Object Gateway
  - `mcg-operator`

Administrators define the desired end state of the cluster, and the Fusion Data Foundation operators ensure the cluster is either in that state or approaching that state, with minimal administrator intervention.

- **Fusion Data Foundation operator**
  The `odf-operator` can be described as a "meta" operator for Fusion Data Foundation, that is, an operator meant to influence other operators.
- **Container Storage operator**
  The `ocs-operator` can be described as a "meta" operator for Fusion Data Foundation, that is, an operator meant to influence other operators and serves as a configuration gateway for the features provided by the other operators. It does not directly manage the other operators.
- **Rook-Ceph operator**
  Rook-Ceph operator is the Rook operator for Ceph in the Fusion Data Foundation. Rook enables Ceph storage clusters to run on the OpenShift Container Platform.
- **MCG operator**
  The Multicloud Object Gateway (MCG) operator is an operator for Fusion Data Foundation along with the Fusion Data Foundation operator and the Rook-Ceph operator. The MCG operator is available upstream as a standalone operator.

Fusion Data Foundation operator

The `odf-operator` can be described as a "meta" operator for Fusion Data Foundation, that is, an operator meant to influence other operators.

The `odf-operator` has the following primary functions:

- Enforces the configuration and versioning of the other operators that comprise Fusion Data Foundation. It does this by using two primary mechanisms: operator dependencies and Subscription management.
  - The `odf-operator` bundle specifies dependencies on other OLM operators to make sure they are always installed at specific versions.
  - The operator itself manages the Subscriptions for all other operators to make sure the desired versions of those operators are available for installation by the OLM.
- Provides the Fusion Data Foundation external plugin for the OpenShift Console.
- Provides an API to integrate storage solutions with the OpenShift Console.

- **Components**
  The `odf-operator` has a dependency on the `ocs-operator` package. It also manages the Subscription of the `mcg-operator`. In addition, the `odf-operator` bundle defines a second Deployment for the Fusion Data Foundation external plugin for the OpenShift Console.
- **Design diagram**
  This diagram illustrates how `odf-operator` is integrated with the Red Hat® OpenShift® Container Platform.
- **Responsibilites**
  The `odf-operator` defines `StorageSystem` CRD.
- **Resources**
  The `ocs-operator` creates resources in response to the spec of a given StorageSystem.
- **Limitation**
  The `odf-operator` does not provide any data storage or services itself. It exists as an integration and management layer for other storage systems.
- **High availability**
  High availability is not a primary requirement for the `odf-operator` Pod similar to most of the other operators. In general, there are no operations that require or benefit from process distribution.
- **Relevant config files**
  The `odf-operator` comes with a `ConfigMap` of variables that can be used to modify the behavior of the operator.
- **Relevant log files**
  Use the relevant log files to understand Fusion Data Foundation and troubleshoot issues.
- **Lifecycle**
  The `odf-operator` is required to be present as long as the Fusion Data Foundation bundle remains installed. This is managed as part of OLM's reconciliation of the Fusion Data Foundation CSV. At least one instance of the pod should be in `Ready` state.

Components

The `odf-operator` has a dependency on the `ocs-operator` package. It also manages the Subscription of the `mcg-operator`. In addition, the `odf-operator` bundle defines a second Deployment for the Fusion Data Foundation external plugin for the OpenShift Console.

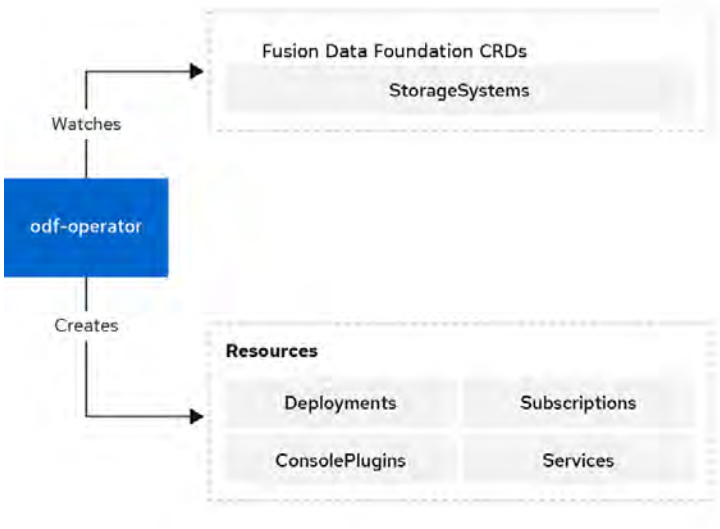
This defines an `nginx`-based Pod that serves the necessary files to register and integrate Fusion Data Foundation dashboards directly into the OpenShift Container Platform Console.

Design diagram

This diagram illustrates how `odf-operator` is integrated with the Red Hat® OpenShift® Container Platform.

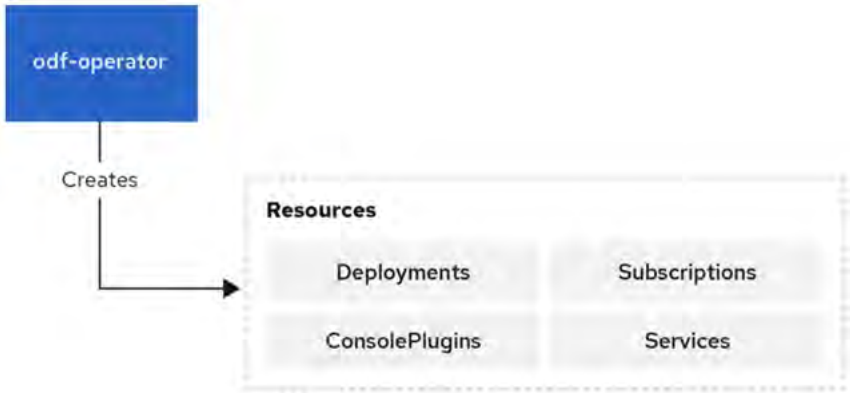
For Fusion Data Foundation 4.18 or earlier releases:

Figure 1. Fusion Data Foundation Operator



For Fusion Data Foundation 4.19 or later releases:

Figure 2. Fusion Data Foundation Operator



Responsibilites

The **odf-operator** defines **StorageSystem** CRD.

The **StorageSystem** CRD represents an underlying storage system that provides data storage and services for OpenShift Container Platform. It triggers the operator to ensure the existence of a **Subscription** for a given **Kind** of storage system.

Resources

The **ocs-operator** creates resources in response to the spec of a given StorageSystem.

Operator Lifecycle Manager Resources
: Creates a **Subscription** for the operator which defines and reconciles the given **StorageSystem**'s **Kind**.

Limitation

The **odf-operator** does not provide any data storage or services itself. It exists as an integration and management layer for other storage systems.

High availability

High availability is not a primary requirement for the **odf-operator** Pod similar to most of the other operators. In general, there are no operations that require or benefit from process distribution.

OpenShift Container Platform quickly spins up a replacement Pod whenever the current Pod becomes unavailable or is deleted.

Relevant config files

The `odf-operator` comes with a `ConfigMap` of variables that can be used to modify the behavior of the operator.

Relevant log files

Use the relevant log files to understand Fusion Data Foundation and troubleshoot issues.

Use the following to understand and troubleshoot Fusion Data Foundation:

Operator Pod logs
: Each operator provides standard Pod logs that include information about reconciliation and errors encountered. These logs often have information about successful reconciliation which can be filtered out and ignored.

`StorageSystem` status
: The `StorageSystem` CR stores the reconciliation details in the status of the CR and has associated events. The spec of the `StorageSystem` contains the name, namespace, and `Kind` of the actual storage system's CRD, which the administrator can use to find further information on the status of the storage system.

Underlying storage system CRD statuses

Lifecycle

The `odf-operator` is required to be present as long as the Fusion Data Foundation bundle remains installed. This is managed as part of OLM's reconciliation of the Fusion Data Foundation CSV. At least one instance of the pod should be in `Ready` state.

The operator operands such as CRDs should not affect the lifecycle of the operator. The creation and deletion of `StorageSystems` is an operation outside the operator's control and must be initiated by the administrator or automated with the appropriate application programming interface (API) calls.

Container Storage operator

The `ocs-operator` can be described as a "meta" operator for Fusion Data Foundation, that is, an operator meant to influence other operators and serves as a configuration gateway for the features provided by the other operators. It does not directly manage the other operators.

The `ocs-operator` has the following primary functions:

- Creates Custom Resources (CRs) that trigger the other operators to reconcile against them.
- Abstracts the Ceph and Multicloud Object Gateway configurations and limits them to known best practices that are validated and supported by Red Hat.
- Creates and reconciles the resources required to deploy containerized Ceph and NooBaa according to the support policies.

- **Components**
  The `ocs-operator` does not have any dependent components. However, the operator has a dependency on the existence of all the custom resource definitions (CRDs) from other operators, which are defined in the `ClusterServiceVersion` (CSV).
- **Design diagram**
  This diagram illustrates how Fusion Data Foundation is integrated with the OpenShift Container Platform.
- **Responsibilities**
  The `ocs-operator` defines the associated CRDs.
- **Resources**
  The `ocs-operator` creates the following CRs in response to the spec of the CRDs it defines.
- **Limitation**
  The `ocs-operator` neither deploys nor reconciles the other Pods of Fusion Data Foundation. The `ocs-operator` CSV defines the top-level components such as operator Deployments and the Operator Lifecycle Manager (OLM) reconciles the specified component.
- **High availability**
  High availability is not a primary requirement for the `ocs-operator` Pod similar to most of the other operators. In general, there are no operations that require or benefit from process distribution.
- **Relevant config files**
  The `ocs-operator` configuration is entirely specified by the CSV and is not modifiable without a custom build of the CSV.
- **Relevant log files**
  Use the relevant log files to understand Fusion Data Foundation and troubleshoot issues.
- **Lifecycle**
  The `ocs-operator` is required to be present as long as the container storage bundle remains installed. This is managed as part of OLM's reconciliation of the Container Storage CSV. At least one instance of the pod should be in `Ready` state.

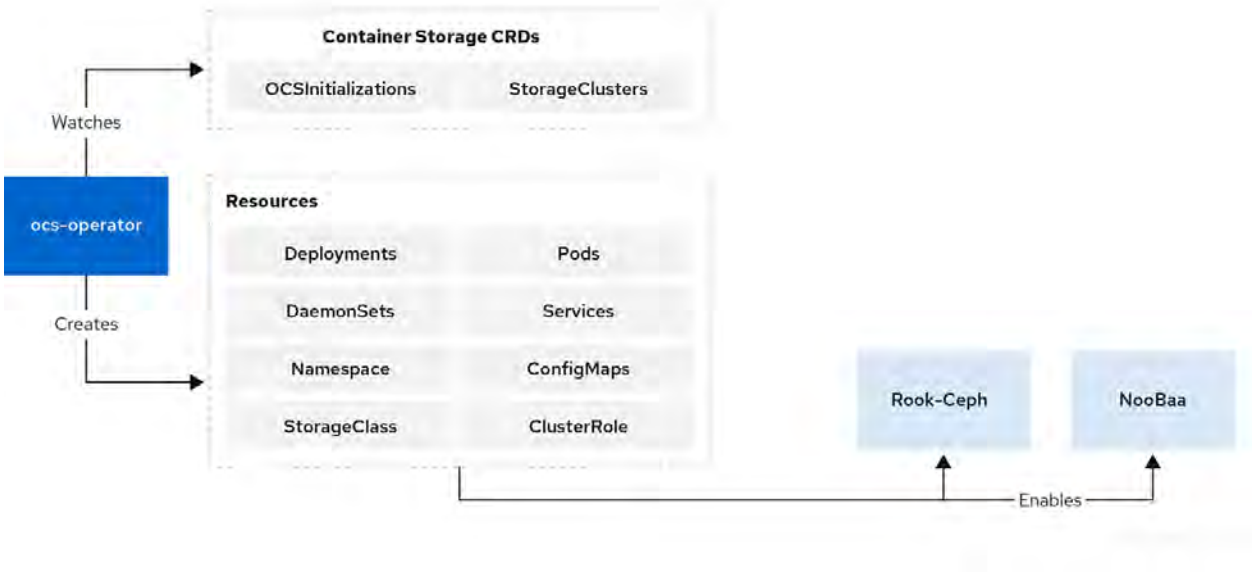
Components

The `ocs-operator` does not have any dependent components. However, the operator has a dependency on the existence of all the custom resource definitions (CRDs) from other operators, which are defined in the `ClusterServiceVersion` (CSV).

Design diagram

This diagram illustrates how Fusion Data Foundation is integrated with the OpenShift Container Platform.

Figure 1. Container Storage Operator



Responsibilities

The `ocs-operator` defines the associated CRDs.

The two `ocs-operator` CRDs are:

`OCSInitialization`
: `OCSInitialization` is a singleton CRD used for encapsulating operations that apply at the operator level. The operator takes care of ensuring that one instance always exists. The CR triggers the following:

  - Performs initialization tasks required for Fusion Data Foundation. If needed, these tasks can be triggered to run again by deleting the `OCSInitialization` CRD.
    - Ensures that the required Security Context Constraints (SCCs) for Fusion Data Foundation are present.
  - Manages the deployment of the Ceph toolbox Pod, used for performing advanced troubleshooting and recovery operations.

`StorageCluster`
: The `StorageCluster` CRD represents the system that provides the full functionality of Fusion Data Foundation. It triggers the operator to ensure the generation and reconciliation of `Rook-Ceph` and `NooBaa` CRDs. The `ocs-operator` algorithmically generates the `CephCluster` and `NooBaa` CRDs based on the configuration in the `StorageCluster` spec. The operator also creates additional CRs, such as `CephBlockPools`, `Routes`, and so on. These resources are required for enabling different features of Fusion Data Foundation. Currently, only one StorageCluster CR per OpenShift Container Platform cluster is supported.

Resources

The `ocs-operator` creates the following CRs in response to the spec of the CRDs it defines.

The configuration of some of these resources can be overridden, allowing for changes to the generated spec or not creating them altogether.

- General resources
- Rook-Ceph resources
- Multicloud Object Gateway resources
- Monitoring resources

General resources

Events
: Creates various events when required in response to reconciliation.

Persistent Volumes (PVs)
: PVs are not created directly by the operator. However, the operator keeps track of all the PVs created by the Ceph CSI drivers and ensures that the PVs have appropriate annotations for the supported features.

Quickstarts
: Deploys various Quickstart CRs for the OpenShift Container Platform Console.

Rook-Ceph resources

`CephBlockPool`

Define the default Ceph block pools.

CephFilesysPrometheusRulesoute for the Ceph object store.

StorageClass
Define the default Storage classes. For example, for **CephBlockPool** and **CephFilesystem**).

VolumeSnapshotClass
Define the default volume snapshot classes for the corresponding storage classes.

Multicloud Object Gateway resources

NooBaa
Define the default Multicloud Object Gateway system.

Monitoring resources

- Metrics Exporter Service
- Metrics Exporter Service Monitor
- PrometheusRules

Limitation

The **ocs-operator** neither deploys nor reconciles the other Pods of Fusion Data Foundation. The **ocs-operator** CSV defines the top-level components such as operator Deployments and the Operator Lifecycle Manager (OLM) reconciles the specified component.

High availability

High availability is not a primary requirement for the **ocs-operator** Pod similar to most of the other operators. In general, there are no operations that require or benefit from process distribution.

OpenShift Container Platform quickly spins up a replacement Pod whenever the current Pod becomes unavailable or is deleted.

Relevant config files

The **ocs-operator** configuration is entirely specified by the CSV and is not modifiable without a custom build of the CSV.

Relevant log files

Use the relevant log files to understand Fusion Data Foundation and troubleshoot issues.

See the following log and status types:

Operator Pod logs
Each operator provides standard Pod logs that include information about reconciliation and errors encountered. These logs often have information about successful reconciliation which can be filtered out and ignored.

StorageCluster status and events
The **StorageCluster** CR stores the reconciliation details in the status of the CR and has associated events. Status contains a section of the expected container images. It shows the container images that it expects to be present in the pods from other operators and the images that it currently detects. This helps to determine whether the Fusion Data Foundation upgrade is complete.

OCSInitialization status
This status shows whether the initialization tasks are completed successfully.

Lifecycle

The **ocs-operator** is required to be present as long as the container storage bundle remains installed. This is managed as part of OLM's reconciliation of the Container Storage CSV. At least one instance of the pod should be in **Ready** state.

The operator operands such as CRDs should not affect the lifecycle of the operator. An **OCSInitialization** CR should always exist. The operator creates one if it does not exist. The creation and deletion of StorageClusters is an operation outside the operator's control and must be initiated by the administrator or automated with the appropriate API calls.

Rook-Ceph operator

Rook-Ceph operator is the Rook operator for Ceph in the Fusion Data Foundation. Rook enables Ceph storage clusters to run on the OpenShift Container Platform.

The Rook-Ceph operator is a simple container that automatically bootstraps the storage clusters and monitors the storage daemons to ensure the storage clusters are healthy.

- **Components**
  The Rook-Ceph operator manages a number of components as part of the Fusion Data Foundation deployment.
- **Design diagram**
  With Ceph running in the OpenShift Container Platform cluster, OpenShift Container Platform applications can mount block devices and filesystems managed by Rook-Ceph, or can use the S3/Swift API for object storage.
- **Responsibilities**
  The Rook-Ceph operator is a container that bootstraps and monitors the storage cluster, performing various functions.
- **Resources**
  Rook-Ceph operator adds owner references to all the resources it creates in the `openshift-storage` namespace.
- **Lifecycle**
  Rook-Ceph operator manages the lifecycle of pods in the Ceph cluster.

Components

The Rook-Ceph operator manages a number of components as part of the Fusion Data Foundation deployment.

Ceph-CSI Driver
: The operator creates and updates the CSI driver, including a provisioner for each of the two drivers, RADOS block device (RBD) and Ceph filesystem (CephFS) and a volume plugin `daemonset` for each of the two drivers.

Ceph daemons

Mons
: The monitors (mons) provide the core metadata store for Ceph.

OSDs
: The object storage daemons (OSDs) store the data on underlying devices.

Mgr
: The manager (mgr) collects metrics and provides other internal functions for Ceph.

RGW
: The RADOS Gateway (RGW) provides the S3 endpoint to the object store.

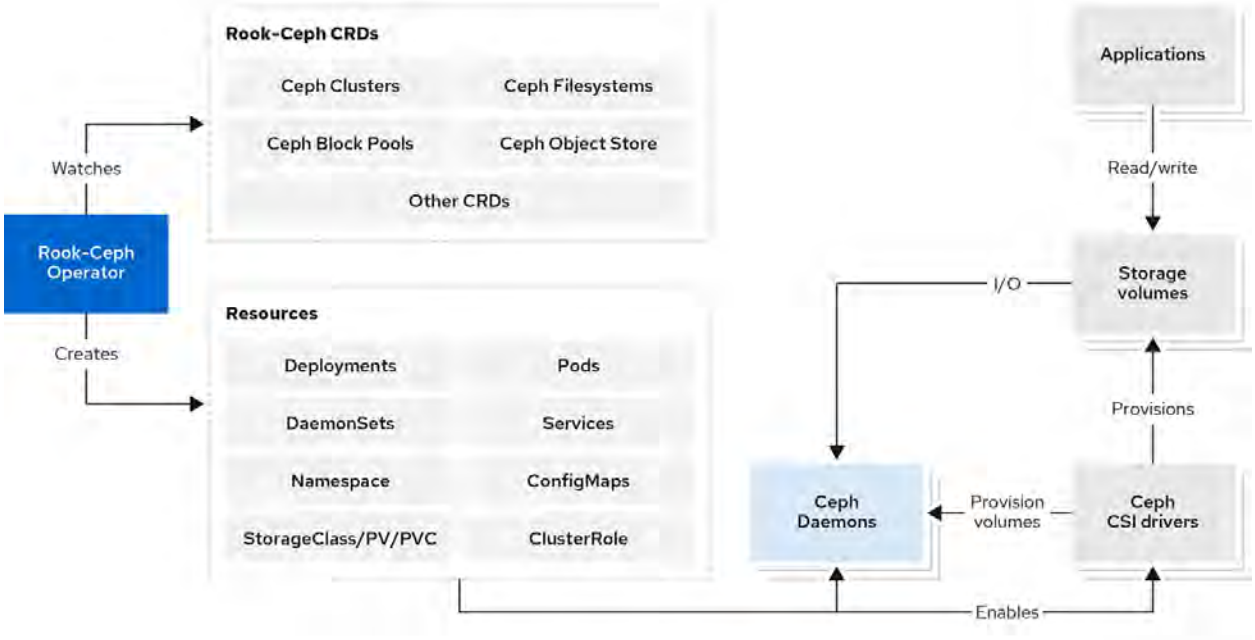
MDS
: The metadata server (MDS) provides CephFS shared volumes.

Design diagram

With Ceph running in the OpenShift Container Platform cluster, OpenShift Container Platform applications can mount block devices and filesystems managed by Rook-Ceph, or can use the S3/Swift API for object storage.

Figure 1 illustrates how Ceph Rook integrates with OpenShift Container Platform.

Figure 1. Rook-Ceph Operator



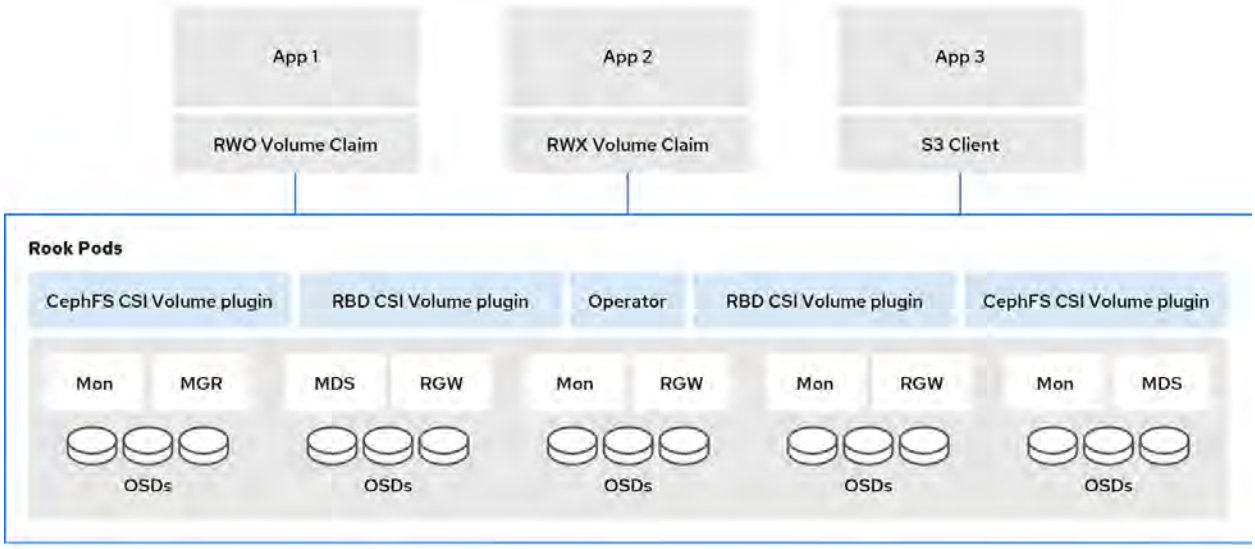
Responsibilities

The Rook-Ceph operator is a container that bootstraps and monitors the storage cluster, performing various functions.

These functions include the following:

- Automates the configuration of storage components
- Starts, monitors, and manages the Ceph monitor pods and Ceph OSD daemons to provide the RADOS storage cluster
- Initializes the pods and other artifacts to run the services to manage:
    - CRDs for pools
    - Object stores (S3/Swift)
    - Filesystems
- Monitors the Ceph mons and OSDs to ensure that the storage remains available and healthy
- Deploys and manages Ceph mons placement while adjusting the mon configuration based on cluster size
- Watches the desired state changes requested by the API service and applies the changes
- Initializes the Ceph-CSI drivers that are needed for consuming the storage
- Automatically configures the Ceph-CSI driver to mount the storage to pods

Figure 1. Rook-Ceph Operator architecture



The Rook-Ceph operator image includes all required tools to manage the cluster. There is no change to the data path. However, the operator does not expose all Ceph configurations. Many of the Ceph features like placement groups and crush maps are hidden from the users and are provided with a better user experience in terms of physical resources, pools, volumes, filesystems, and buckets.

Resources

Rook-Ceph operator adds owner references to all the resources it creates in the **`openshift-storage`** namespace.

When the cluster is uninstalled, the owner references ensure that the resources are all cleaned up. This includes OpenShift Container Platform resources such as **`configmaps`**, **`secrets`**, **`services`**, **`deployments`**, **`daemonsets`**, and so on.

The Rook-Ceph operator watches CRs to configure the settings determined by Fusion Data Foundation, which includes **`CephCluster`**, **`CephObjectStore`**, **`CephFilesystem`**, and **`CephBlockPool`**.

Lifecycle

Rook-Ceph operator manages the lifecycle of pods in the Ceph cluster.

The pods that are managed in the Ceph cluster are as follows:

Rook operator
: A single pod that owns the reconcile of the cluster.

RBD CSI Driver

- Two nodeplugin pods, managed by a single deployment.
- One ctrlplugin pod per node, managed by a **`daemonset`**.

CephFS CSI Driver

- Two nodeplugin pods, managed by a single deployment.
- One ctrlplugin pod per node, managed by a **`daemonset`**.

Monitors (mons)
: Three mon pods, each with its own deployment.

  Stretch clusters
  : Contain five mon pods, one in the arbiter zone and two in each of the other two data zones.

Manager (mgr)
: There is a single mgr pod for the cluster.

  Stretch clusters
  : There are two mgr pods (starting with Fusion Data Foundation 4.8), one in each of the two non-arbiter zones.

Object storage daemons (OSDs)
: At least three OSDs are created initially in the cluster. More OSDs are added when the cluster is expanded.

Metadata server (MDS)

The CephFS metadata server has a single pod.

RADOS gateway (RGW)
: The Ceph RGW daemon has a single pod.

MCG operator

The Multicloud Object Gateway (MCG) operator is an operator for Fusion Data Foundation along with the Fusion Data Foundation operator and the Rook-Ceph operator. The MCG operator is available upstream as a standalone operator.

The MCG operator performs the following primary functions:

- Controls and reconciles the Multicloud Object Gateway (MCG) component within Fusion Data Foundation.
- Manages new user resources such as object bucket claims, bucket classes, and backing stores.
- Creates the default out-of-the-box resources.

A few configurations and information are passed to the MCG operator through the Fusion Data Foundation operator.

- **Components**
  The MCG operator does not have sub-components. However, it consists of a reconcile loop for the different resources that are controlled by it.
- **Responsibilities and resources**
  The MCG operator reconciles and is responsible for the custom resource definitions (CRDs) and OpenShift Container Platform entities.
- **High availability**
  As an operator, the only high availability provided is that the OpenShift Container Platform reschedules a failed pod.
- **Relevant log files**
  Use the relevant log files to help troubleshoot issues with the NooBaa operator.
- **Lifecycle**
  The MCG operator runs and reconciles after Fusion Data Foundation is deployed and until it is uninstalled.

Components

The MCG operator does not have sub-components. However, it consists of a reconcile loop for the different resources that are controlled by it.

The MCG operator has a command-line interface (CLI) and is available as a part of Fusion Data Foundation. It enables the creation, deletion, and querying of various resources. This CLI adds a layer of input sanitation and status validation before the configurations are applied unlike applying a YAML file directly.

Responsibilities and resources

The MCG operator reconciles and is responsible for the custom resource definitions (CRDs) and OpenShift Container Platform entities.

- Backing store
- Namespace store
- Bucketclass
- Object bucket claims (OBCs)
- NooBaa, pod stateful sets CRD
- DB pods
- Prometheus rules and service monitoring
- Horizontal pod autoscaler (HPA)

Backing store

A resource that the customer has connected to the MCG component. This resource provides MCG the ability to save the data of the provisioned buckets on top of it.

A default backing store is created as part of the deployment depending on the platform that the OpenShift Container Platform is running on. For example, when OpenShift Container Platform or Fusion Data Foundation is deployed on Amazon Web Services (AWS), it results in a default backing store which is an AWS::S3 bucket. Similarly, for Microsoft Azure, the default backing store is a blob container and so on.

The default backing stores are created using CRDs for the cloud credential operator, which comes with OpenShift Container Platform. There is no limit on the amount of the backing stores that can be added to MCG. The backing stores are used in the bucket class CRD to define the different policies of the bucket. Refer the documentation of the specific Fusion Data Foundation version to identify the types of services or resources supported as backing stores.

Namespace store

Resources that are used in namespace buckets. No default is created during deployment.

Bucketclass

A default or initial policy for a newly provisioned bucket. The following policies are set in a bucketclass:

Placement policy

Indicates the backing stores to be attached to the bucket and used to write the data of the bucket. This policy is used for data buckets and for cache policies to indicate the local cache placement. There are two modes of placement policy:

- Spread. Strips the data across the defined backing stores
- Mirror. Creates a full replica on each backing store

Namespace policy
: A policy for the namespace buckets that defines the resources that are being used for aggregation and the resource used for the write target.

Cache Policy
: This is a policy for the bucket and sets the hub (the source of truth) and the time to live (TTL) for the cache items.

A default bucket class is created during deployment and it is set with a placement policy that uses the default backing store. There is no limit to the number of bucket class that can be added.

Refer to the documentation of the specific Fusion Data Foundation version to identify the types of policies that are supported.

Object bucket claims (OBCs)

CRDs that enable provisioning of S3 buckets. With MCG, OBCs receive an optional bucket class to note the initial configuration of the bucket. If a bucket class is not provided, the default bucket class is used.

NooBaa, pod stateful sets CRD

An internal CRD that controls the different pods of the NooBaa deployment such as the DB pod, the core pod, and the endpoints. This CRD must not be changed as it is internal. This operator reconciles the following entities:

- DB pod SCC
- Role Binding and Service Account to allow SSO single sign-on between OpenShift Container Platform and NooBaa user interfaces
- Route for S3 access
- Certificates that are taken and signed by the OpenShift Container Platform and are set on the S3 route

DB pods

The MCG DB pods serve as the storage layer for MCG metadata. They are configured to enable high availability (HA) with a primary-secondary architecture. The primary pod handles write operations and most reads, while the secondary pod maintains a synchronized replica for redundancy and automatic failover. The secondary pod also handles certain read-only operations, reducing the load from the primary pod.

Prometheus rules and service monitoring

These CRDs set up scraping points for Prometheus and alert rules that are supported by MCG.

Horizontal pod autoscaler (HPA)

It is Integrated with the MCG endpoints. The endpoint pods scale up and down according to CPU pressure (amount of S3 traffic).

High availability

As an operator, the only high availability provided is that the OpenShift Container Platform reschedules a failed pod.

Relevant log files

Use the relevant log files to help troubleshoot issues with the NooBaa operator.

To troubleshoot issues with the NooBaa operator, you can look at the following:

- Operator pod logs, which are also available through the must-gather.
- Different CRDs or entities and their statuses that are available through the must-gather.

Lifecycle

The MCG operator runs and reconciles after Fusion Data Foundation is deployed and until it is uninstalled.

Fusion Data Foundation installation overview

Fusion Data Foundation consists of multiple components managed by multiple operators.

- **Installed Operators**
  When installing Fusion Data Foundation from the Operator Hub, four separate deployments are created. These operators run independently and interact with each other by creating customer resources (CRs) that are watched by the other operators.
- **OCSInitialization CR**
  The Fusion Data Foundation bundle defines an external plugin to the OpenShift Container Platform Console, adding new screens and functionality not otherwise available in the Console. This plugin runs as a web server in the `odf-console-plugin` Pod, which is managed by a Deployment created by the OLM at the time of installation.
- **Storage cluster creation**
  The Fusion Data Foundation operators themselves provide no storage functionality, and the desired storage configuration must be defined.

Installed Operators

When installing Fusion Data Foundation from the Operator Hub, four separate deployments are created. These operators run independently and interact with each other by creating customer resources (CRs) that are watched by the other operators.

The four separate deployments created are as follows:

`odf-operator`
: Defines the `odf-operator` Pod.

`ocs-operator`
: Defines the `ocs-operator` Pod which runs processes for `ocs-operator` and its `metrics-exporter` in the same container.
The `ocs-operator` is primarily responsible for creating the CRs to configure Ceph storage and Multicloud Object Gateway.

`rook-ceph-operator`
: Defines the `rook-ceph-operator` Pod.

`mcg-operator`
: Defines the `mcg-operator` Pod.
The `mcg-operator` sometimes creates Ceph volumes for use by its components.

OCSInitialization CR

The Fusion Data Foundation bundle defines an external plugin to the OpenShift Container Platform Console, adding new screens and functionality not otherwise available in the Console. This plugin runs as a web server in the `odf-console-plugin` Pod, which is managed by a Deployment created by the OLM at the time of installation.

The `ocs-operator` automatically creates an `OCSInitialization` CR after it gets created. Only one `OCSInitialization` CR exists at any point in time. It controls the `ocs-operator` behaviors that are not restricted to the scope of a single `StorageCluster`, but only performs them once. When you delete the `OCSInitialization` CR, the `ocs-operator` creates it again and this you can re-trigger its initialization operations.

The `OCSInitialization` CR controls the following behaviors:

`SecurityContextConstraints` **(SCCs)**
: After the `OCSInitialization` CR is created, the `ocs-operator` creates various SCCs for use by the component Pods.

Ceph Toolbox Deployment
: Use the `OCSInitialization` to deploy the Ceph Toolbox Pod for the advanced Ceph operations.

Rook-Ceph Operator Configuration
: This configuration creates the `rook-ceph-operator-configConfigMap` that governs the overall configuration for `rook-ceph-operator` behavior.

Storage cluster creation

The Fusion Data Foundation operators themselves provide no storage functionality, and the desired storage configuration must be defined.

After you install the operators, create a new `StorageCluster`, using either the Fusion Data Foundation console wizard or the CLI and the `ocs-operator` reconciles this `StorageCluster`. Fusion Data Foundation supports a single `StorageCluster` per installation. Any `StorageCluster` CRs created after the first one is ignored by `ocs-operator` reconciliation.

Fusion Data Foundation allows the following StorageCluster configurations:

Internal
: In the Internal mode, all the components run containerized within the Fusion Data Foundation cluster and uses dynamically provisioned persistent volumes (PVs) created against the `StorageClass` specified by the administrator in the installation wizard.

Internal-attached
: This mode is similar to the Internal mode but the administrator is required to define the local storage devices directly attached to the cluster nodes that the Ceph uses for its backing storage. Also, the administrator need to create the CRs that the local storage operator reconciles to provide the `StorageClass`. The `ocs-operator` uses this `StorageClass` as the backing storage for Ceph.

External
: In this mode, Ceph components do not run inside the Fusion Data Foundation cluster instead connectivity is provided to an external Fusion Data Foundation installation for which the applications can create PVs. The other components run within the cluster as required.

MCG Standalone
: This mode facilitates the installation of a Multicloud Object Gateway system without an accompanying CephCluster.

After a `StorageCluster` CR is found, `ocs-operator` validates it and begins to create subsequent resources to define the storage components.

- **Internal mode storage cluster**
  Both internal and internal-attached storage clusters follow the same setup process, deploying Ceph OSDs, the CephFS filesystem, and RBD block pools on locally attached storage nodes.
- **External mode storage cluster**
  For external storage clusters, `ocs-operator` follows a slightly different setup process. The `ocs-operator` looks for the existence of the `rook-ceph-external-cluster-details` `ConfigMap`, which must be created by someone else, either the administrator or the Console.
- **Standalone Multicloud Object Gateway storage cluster**
  In Standalone Multicloud Object Gateway mode, no CephCluster is created. Instead, a NooBaa system custom resource (CR) is created using default values to take advantage of the preexisting storage classes in the OpenShift Container Platform dashboards.

Internal mode storage cluster

Both internal and internal-attached storage clusters follow the same setup process, deploying Ceph OSDs, the CephFS filesystem, and RBD block pools on locally attached storage nodes.

Both internal and internal-attached storage clusters have the same setup process as follows:

`StorageClasses`	Create the storage classes that cluster applications use to create Ceph volumes.
`SnapshotClasses`	Create the volume snapshot classes that the cluster applications use to create snapshots of Ceph volumes.
Ceph RGW configuration	Create various Ceph object CRs to enable and provide access to the Ceph RGW object storage endpoint.
Ceph RBD Configuration	Create the `CephBlockPool` CR to enable RBD storage.
CephFS Configuration	Create the `CephFilesystem` CR to enable CephFS storage.
Rook-Ceph Configuration	Create the `rook-config-override` `ConfigMap` that governs the overall behavior of the underlying Ceph cluster.
`CephCluster`	Create the `CephCluster` CR to trigger Ceph reconciliation from `rook-ceph-operator`. For more information, see Rook-Ceph operator.
`NoobaaSystem`	Create the `NooBaa` CR to trigger reconciliation from `mcg-operator`. For more information, see MCG operator.
Job templates	Create `OpenShift Template` CRs that define Jobs to run administrative operations for Fusion Data Foundation.
Quickstarts	Create the `QuickStart` CRs that display the quickstart guides in the Web Console.

- **Cluster creation**
  The `rook-operator` and Rook complete the cluster component configuration and setup.
- **NooBaa System creation**
  When a NooBaa system is created, the `mcg-operator` reconciles a default BackingStore, a default BucketClass, and an S3 endpoint to enable object storage access through the Multicloud Object Gateway.

Cluster creation

The `rook-operator` and Rook complete the cluster component configuration and setup.

After the `ocs-operator` creates the `CephCluster` CR, the `rook-operator` creates the Ceph cluster according to the desired configuration.

The `rook-operator` configures the various components, as detailed in Table 1.

Table 1. `rook-operator` component configurations

Component	Description
Ceph `mon` daemons	Three Ceph `mon` daemons are started on different nodes in the cluster. They manage the core metadata for the Ceph cluster and they must form a majority quorum. The metadata for each `mon` is backed either by a PV if it is in a cloud environment or a path on the local host if it is in a local storage device environment.
Ceph `mgr` daemon	This daemon is started and it gathers metrics for the cluster and report them to Prometheus.
Ceph OSDs	These OSDs are created according to the configuration of the `storageClassDeviceSets`. Each OSD consumes a PV that stores the user data. By default, Ceph maintains three replicas of the application data across different OSDs for high durability and availability using the `CRUSH` algorithm.
CSI provisioners	These provisioners are started for RBD and `CephFS`. When volumes are requested for the storage classes of Fusion Data Foundation, the requests are directed to the `Ceph-CSI` driver to provision the volumes in Ceph.
CSI volume plugins and `CephFS`	The CSI volume plugins for RBD and `CephFS` are started on each node in the cluster. The volume plugin needs to be running wherever the Ceph volumes are required to be mounted by the applications.

After the `CephCluster` CR is configured, Rook reconciles the remaining Ceph CRs to complete the setup, as detailed in Table 2.

Table 2. Ceph customer resources

Ceph CRs	Description
`CephBlockPool`	The `CephBlockPool` CR provides the configuration for Rook operator to create Ceph pools for RWO volumes.
`CephFilesystem`	The `CephFilesystem` CR instructs the Rook operator to configure a shared file system with CephFS, typically for RWX volumes. The CephFS metadata server (MDS) is started to manage the shared volumes.
`CephObjectStore`	The `CephObjectStore` CR instructs the Rook operator to configure an object store with the RGW service.
`CephObjectStore User` CR	The `CephObjectStoreUser` CR instructs the Rook operator to configure an object store user for NooBaa to consume, publishing `access/private` key as well as the `CephObjectStore` endpoint.

The operator monitors the Ceph health to ensure that storage platform remains healthy. If a `mon` daemon goes down for too long a period (10 minutes), Rook starts a new `mon` in its place so that the full quorum can be fully restored.

When the `ocs-operator` updates the `CephCluster` CR, Rook immediately responds to the requested changes to update the cluster configuration.

NooBaa System creation

When a NooBaa system is created, the **mcg-operator** reconciles a default BackingStore, a default BucketClass, and an S3 endpoint to enable object storage access through the Multicloud Object Gateway.

When a NooBaa system is created, the **mcg-operator** reconciles the following:

Default BackingStore

Depending on the platform that OpenShift Container Platform and Fusion Data Foundation are deployed on, a default backing store resource is created so that buckets can use it for their placement policy. The different options are as follows:

Amazon Web Services (AWS) deployment	The **mcg-operator** uses the **CloudCredentialsOperator** (CCO) to mint credentials in order to create a new AWS::S3 bucket and creates a **BackingStore** on top of that bucket.
Microsoft Azure deployment	The **mcg-operator** uses the CCO to mint credentials in order to create a new Azure Blob and creates a **BackingStore** on top of that bucket.
Google Cloud Platform (GCP) deployment	The **mcg-operator** uses the CCO to mint credentials in order to create a new GCP bucket and creates a BackingStore on top of that bucket.
On-prem deployment	If RGW exists, the **mcg-operator** creates a new **CephUser** and a new bucket on top of RGW and create a **BackingStore** on top of that bucket.
None of the previously mentioned deployments are applicable	The **mcg-operator** creates a **pv-pool** based on the default storage class and creates a **BackingStore** on top of that bucket.

Default `BucketClass`

A **BucketClass** with a placement policy to the default **BackingStore** is created.

NooBaa pods

The following NooBaa pods are created and started:

Database (DB)	This is a Postgres DB holding metadata, statistics, events, and so on. However, it does not hold the actual data being stored.
Core	This is the pod that handles configuration, background processes, metadata management, statistics, and so on.
Endpoints	These pods perform the actual I/O-related work such as deduplication and compression, communicating with different services to write and read data, and so on. The endpoints are integrated with the **HorizonalPodAutoscaler** and their number increases and decreases according to the CPU usage observed on the existing endpoint pods.

Route

A Route for the NooBaa S3 interface is created for applications that uses S3.

Service

A Service for the NooBaa S3 interface is created for applications that uses S3.

External mode storage cluster

For external storage clusters, **ocs-operator** follows a slightly different setup process. The **ocs-operator** looks for the existence of the **rook-ceph-external-cluster-details ConfigMap**, which must be created by someone else, either the administrator or the Console.

For information about how to create the **ConfigMap**, see Deploying Data Foundation in external mode. The **ocs-operator** then creates some or all of the following resources, as specified in the **ConfigMap**:
Important: Be sure to create IBM Fusion Data Foundation in external mode before Fusion deployment.

External Ceph Configuration	A ConfigMap that specifies the endpoints of the external **mons**.
External Ceph Credentials Secret	A Secret that contains the credentials to connect to the external Ceph instance.
External Ceph StorageClasses	One or more StorageClasses to enable the creation of volumes for RBD, CephFS, and/or RGW.
Enable CephFS CSI Driver	If a **CephFS** StorageClass is specified, configure **rook-ceph-operator** to deploy the **CephFS** CSI Pods.
Ceph RGW Configuration	If an RGW StorageClass is specified, create various Ceph Object CRs to enable and provide access to the Ceph RGW object storage endpoint.

After creating the resources specified in the **ConfigMap**, the StorageCluster creation process proceeds as follows:

CephCluster	Create the **CephCluster** CR to trigger Ceph reconciliation from **rook-ceph-operator** (see subsequent sections).
SnapshotClasses	Create the **SnapshotClasses** that applications use to create snapshots of Ceph volumes.
NoobaaSystem	Create the **NooBaa** CR to trigger reconciliation from noobaa-operator (see subsequent sections).
QuickStarts	Create the **Quickstart** CRs that display the quickstart guides in the Console.

- **Cluster Creation**
  When a **CephCluster** CR is created in external mode, the Rook operator validates the remote Ceph connection, configures the CSI driver, and creates the StorageClasses needed to provision persistent volumes.

- **NooBaa System creation**
  When a NooBaa system is created, the `mcg-operator` reconciles a default BackingStore, a default BucketClass, and an S3 endpoint to enable object storage access through the Multicloud Object Gateway.

Cluster Creation

When a `CephCluster` CR is created in external mode, the Rook operator validates the remote Ceph connection, configures the CSI driver, and creates the StorageClasses needed to provision persistent volumes.

The Rook operator performs the following operations when the `CephCluster` CR is created in external mode:

- The operator validates that a connection is available to the remote Ceph cluster. The connection requires `mon` endpoints and secrets to be imported into the local cluster.
- The CSI driver is configured with the remote connection to Ceph. The RBD and `CephFS` provisioners and volume plugins are started similarly to the CSI driver when configured in internal mode, the connection to Ceph happens to be external to the OpenShift cluster.
- Periodically watch for monitor address changes and update the Ceph-CSI configuration accordingly.

NooBaa System creation

When a NooBaa system is created, the `mcg-operator` reconciles a default BackingStore, a default BucketClass, and an S3 endpoint to enable object storage access through the Multicloud Object Gateway.

When a NooBaa system is created, the `mcg-operator` reconciles the following:

Default BackingStore

Depending on the platform that OpenShift Container Platform and Fusion Data Foundation are deployed on, a default backing store resource is created so that buckets can use it for their placement policy. The different options are as follows:

Amazon Web Services (AWS) deployment	The `mcg-operator` uses the `CloudCredentialsOperator` (CCO) to mint credentials in order to create a new AWS::S3 bucket and creates a `BackingStore` on top of that bucket.
Microsoft Azure deployment	The `mcg-operator` uses the CCO to mint credentials in order to create a new Azure Blob and creates a `BackingStore` on top of that bucket.
Google Cloud Platform (GCP) deployment	The `mcg-operator` uses the CCO to mint credentials in order to create a new GCP bucket and creates a BackingStore on top of that bucket.
On-prem deployment	If RGW exists, the `mcg-operator` creates a new `CephUser` and a new bucket on top of RGW and create a `BackingStore` on top of that bucket.
None of the previously mentioned deployments are applicable	The `mcg-operator` creates a `pv-pool` based on the default storage class and creates a `BackingStore` on top of that bucket.

Default `BucketClass`

A `BucketClass` with a placement policy to the default `BackingStore` is created.

NooBaa pods

The following NooBaa pods are created and started:

Database (DB)	This is a Postgres DB holding metadata, statistics, events, and so on. However, it does not hold the actual data being stored.
Core	This is the pod that handles configuration, background processes, metadata management, statistics, and so on.
Endpoints	These pods perform the actual I/O-related work such as deduplication and compression, communicating with different services to write and read data, and so on. The endpoints are integrated with the `HorizonalPodAutoscaler` and their number increases and decreases according to the CPU usage observed on the existing endpoint pods.

Route

A Route for the NooBaa S3 interface is created for applications that uses S3.

Service

A Service for the NooBaa S3 interface is created for applications that uses S3.

Standalone Multicloud Object Gateway storage cluster

In Standalone Multicloud Object Gateway mode, no CephCluster is created. Instead, a NooBaa system custom resource (CR) is created using default values to take advantage of the preexisting storage classes in the OpenShift Container Platform dashboards.

- **NooBaa System creation**
  When a NooBaa system is created, the `mcg-operator` reconciles a default BackingStore, a default BucketClass, and an S3 endpoint to enable object storage access through the Multicloud Object Gateway.
- **Storage Cluster creation**
  As a part of the StorageCluster creation, `odf-operator` automatically creates a corresponding `StorageSystem` CR, which exposes the StorageCluster to the Fusion Data Foundation.

NooBaa System creation

When a NooBaa system is created, the `mcg-operator` reconciles a default BackingStore, a default BucketClass, and an S3 endpoint to enable object storage access through the Multicloud Object Gateway.

When a NooBaa system is created, the `mcg-operator` reconciles the following:

Default BackingStore

Depending on the platform that OpenShift Container Platform and Fusion Data Foundation are deployed on, a default backing store resource is created so that buckets can use it for their placement policy. The different options are as follows:

Amazon Web Services (AWS) deployment	The `mcg-operator` uses the `CloudCredentialsOperator` (CCO) to mint credentials in order to create a new AWS::S3 bucket and creates a `BackingStore` on top of that bucket.
Microsoft Azure deployment	The `mcg-operator` uses the CCO to mint credentials in order to create a new Azure Blob and creates a `BackingStore` on top of that bucket.
Google Cloud Platform (GCP) deployment	The `mcg-operator` uses the CCO to mint credentials in order to create a new GCP bucket and creates a BackingStore on top of that bucket.
On-prem deployment	If RGW exists, the `mcg-operator` creates a new `CephUser` and a new bucket on top of RGW and create a `BackingStore` on top of that bucket.
None of the previously mentioned deployments are applicable	The `mcg-operator` creates a `pv-pool` based on the default storage class and creates a `BackingStore` on top of that bucket.

Default `BucketClass`

A `BucketClass` with a placement policy to the default `BackingStore` is created.

NooBaa pods

The following NooBaa pods are created and started:

Database (DB)	This is a Postgres DB holding metadata, statistics, events, and so on. However, it does not hold the actual data being stored.
Core	This is the pod that handles configuration, background processes, metadata management, statistics, and so on.
Endpoints	These pods perform the actual I/O-related work such as deduplication and compression, communicating with different services to write and read data, and so on. The endpoints are integrated with the `HorizonalPodAutoscaler` and their number increases and decreases according to the CPU usage observed on the existing endpoint pods.

Route

A Route for the NooBaa S3 interface is created for applications that uses S3.

Service

A Service for the NooBaa S3 interface is created for applications that uses S3.

Storage Cluster creation

As a part of the StorageCluster creation, `odf-operator` automatically creates a corresponding `StorageSystem` CR, which exposes the StorageCluster to the Fusion Data Foundation.

For more information, see Configuring Fusion Data Foundation in MCG-only mode.

Fusion Data Foundation upgrade overview

As an operator bundle managed by the Operator Lifecycle Manager (OLM), Fusion Data Foundation leverages its operators to perform high-level tasks of installing and upgrading the product through `ClusterServiceVersion` (CSV) custom resources (CRs).

- **Upgrade Workflows**
  Fusion Data Foundation recognizes two types of upgrades: Z-stream release upgrades and Minor Version release upgrades.
- **ClusterServiceVersion Reconciliation**
  When the OLM detects an approved `InstallPlan`, it begins the process of reconciling the ClusterServiceVersion (CSVs). This is done by updating the operator

resources based on the new spec, verifying the new CSV installs correctly, then deleting the old CSV.

- **Operator Reconciliation**
  The operators ensure that all relevant resources exist in their expected configurations as specified in the user-facing resources (for example, `StorageCluster`).

Upgrade Workflows

Fusion Data Foundation recognizes two types of upgrades: Z-stream release upgrades and Minor Version release upgrades.

While the user interface workflows for these two upgrade paths are not quite the same, the resulting behaviors are fairly similar. The distinctions are as follows:

For Z-stream releases, Fusion Data Foundation publishes a new bundle in the IBM catalog source. The OLM detects this and creates an `InstallPlan` for the new CSV to replace the existing CSV. The Subscription approval strategy, whether Automatic or Manual, determines whether the OLM proceeds with reconciliation or waits for administrator approval.

For Minor Version releases, Fusion Data Foundation also publishes a new bundle in the IBM catalog source. The difference is that this bundle is part of a new channel, and channel upgrades are not automatic. The administrator must explicitly select the new release channel. After this is done, the OLM detects this and creates an `InstallPlan` for the new CSV to replace the existing CSV. Since the channel switch is a manual operation, OLM automatically starts the reconciliation.

From this point onwards, the upgrade processes are identical.

`ClusterServiceVersion` Reconciliation

When the OLM detects an approved `InstallPlan`, it begins the process of reconciling the ClusterServiceVersion (CSVs). This is done by updating the operator resources based on the new spec, verifying the new CSV installs correctly, then deleting the old CSV.

The upgrade process pushes updates to the operator Deployments, which triggers the restart of the operator Pods using the images specified in the new CSV.

Note: While it is possible to make changes to a given CSV and have those changes propagate to the relevant resource, when upgrading to a new CSV all custom changes are lost, as the new CSV is created based on its unaltered spec.

Operator Reconciliation

The operators ensure that all relevant resources exist in their expected configurations as specified in the user-facing resources (for example, `StorageCluster`).

The reconciliation of the Fusion Data Foundation operands proceeds as defined in An overview of Fusion Data Foundation architecture.

Multus architecture for Fusion Data Foundation

Learn about supported processes and configurations for using the OpenShift multi-network plugin (Multus) with Fusion Data Foundation, including requirements, processes, and features.

This section describes processes and configurations that are supported when configuring Fusion Data Foundation to use the OpenShift multi-network plugin (Multus). This architecture provides information about Fusion Data Foundation requirements, processes, and features that are relevant to Multus.

For configurations that deviate from this reference architecture, open a support exception using the Jira ticket link: https://issues.redhat.com/projects/SUPPORTEX.

- **Assumptions**
  The Multus reference architecture requires internal mode deployment with dedicated storage nodes, assumes OpenShift control plane nodes do not run storage-dependent applications, and supports optional VLAN configurations based on environment requirements.
- **Example scenario used**
  The Multus reference architecture example includes four dedicated storage nodes, two OpenShift worker nodes, and three OpenShift control plane nodes.
- **How to choose Multus architecture for a Fusion Data Foundation deployment**
  Multus networking reduces latency, dedicates bandwidth, and isolates storage networks. Configure Multus for public networks, cluster networks, or both, with public network configuration recommended as a starting point.
- **Multus networking requirements**
  Learn about the general, routing, and address space sizing requirements for implementing Multus networking in your deployment.
- **Recommendations for Multus networking**
  This section describes the general and address range recommendations for Multus networking on Fusion Data Foundation deployment.
- **Planning host interface connections**
  Plan network interface connections for OpenShift nodes based on node type and storage network requirements.
- **Multus network configuration**
  Configure Multus networks using `NetworkAttachmentDefinition` objects with `macvlan` CNI plugin and whereabouts IPAM plugin for IPv4 or IPv6 Pod network connections.
- **Preinstallation checklist for Multus configuration**
  Complete these preinstallation tasks before you deploy Multus networking for your storage cluster.
- **Validating and debugging Multus configuration**
  Fusion Data Foundation has a tool to perform basic connectivity tests prior to Fusion Data Foundation StorageCluster deployment. This Multus validation tool is intended to be a self-help tool that returns a list of details that might be incorrect based on the validation steps that pass or fail.
- **Fusion Data Foundation installation after validating Multus configuration**
  After you validate connectivity by using the Multus validation tool, you can deploy the Fusion Data Foundation storage cluster and select the appropriate Multus networks during deployment.

- **Enabling Multus support on an existing Fusion Data Foundation cluster**
  This procedure describes how to configure Multus networking for an existing Fusion Data Foundation deployment.
- **Multus prerequisites validation tool**
  The Multus prerequisites validation tool validates OpenShift configuration, `NetworkAttachmentDefinitions`, and network compatibility before Fusion Data Foundation installation.

Assumptions

The Multus reference architecture requires internal mode deployment with dedicated storage nodes, assumes OpenShift control plane nodes do not run storage-dependent applications, and supports optional VLAN configurations based on environment requirements.

This Multus reference architecture section assumes the following.

- Fusion Data Foundation is deployed in internal mode using dedicated storage nodes.
- OpenShift control plane nodes do not run applications that require Fusion Data Foundation storage.
  This is not required for supportability, but some node selectors and/or tolerations might need to be modified as needed for other cluster layouts.
- VLANs are present. VLANs are optional and neither recommended nor discouraged. However, in many instances VLANs are required depending on the user environment.

Example scenario used

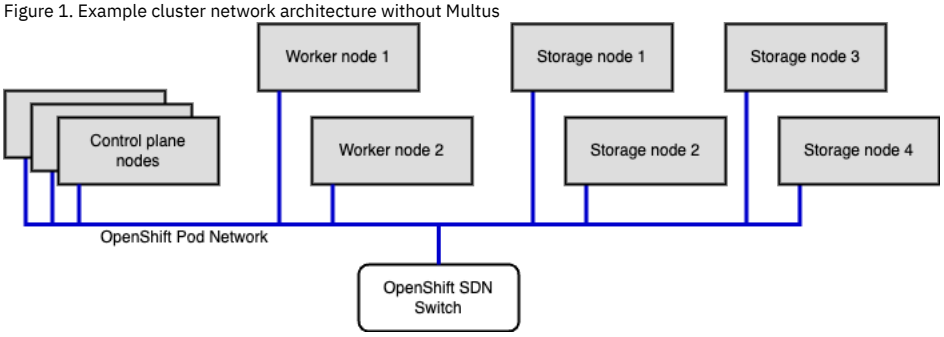
The Multus reference architecture example includes four dedicated storage nodes, two OpenShift worker nodes, and three OpenShift control plane nodes.

This Multus reference architecture uses the following example.

- Four dedicated Fusion Data Foundation storage nodes
- Two OpenShift worker nodes
- Three OpenShift control plane nodes

The following diagram shows the example cluster network architecture in a traditional Fusion Data Foundation deployment without Multus:

Figure 1. Example cluster network architecture without Multus



How to choose Multus architecture for a Fusion Data Foundation deployment

Multus networking reduces latency, dedicates bandwidth, and isolates storage networks. Configure Multus for public networks, cluster networks, or both, with public network configuration recommended as a starting point.

Multus networking for a Fusion Data Foundation deployment is chosen for one or more of the following reasons.

- To reduce latency of the storage system by avoiding the OpenShift Pod (software) network
- To dedicate bandwidth to the storage system to avoid the noisy neighbor problem between storage and applications
- To segregate the storage network from applications for isolation

Multus is enabled for either or both of the following types of Fusion Data Foundation storage networks:

- Public network
- Cluster network

However, it is recommended to configure only the Multus public network as a starting point.

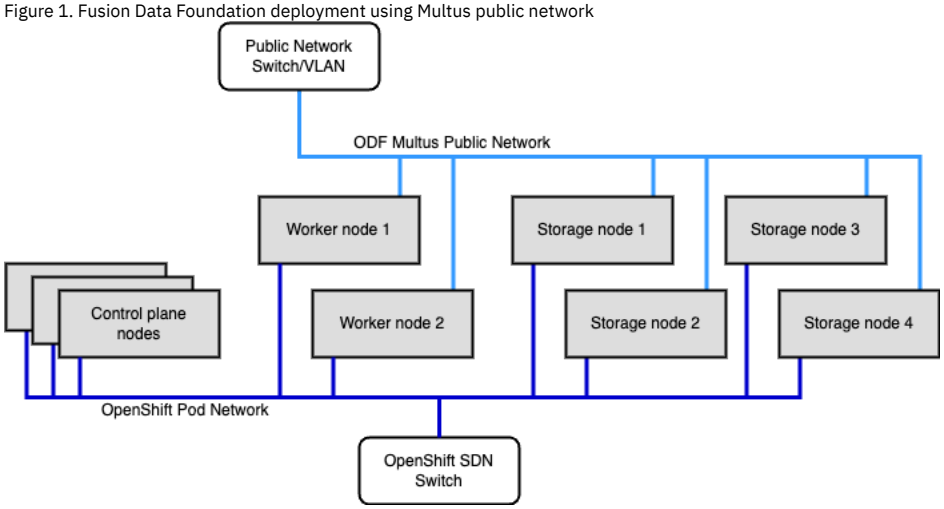
- **Multus public network**
  The public network architecture achieves all three benefits of Multus networking for Fusion Data Foundation deployments.
- **Multus public and cluster networks**
  Using both Multus public and cluster networks simultaneously provides dedicated networks for different types of storage traffic.
- **Multus cluster network**
  The cluster network architecture partially achieves Multus benefits by dedicating a network for replication and recovery traffic.

Multus public network

The public network architecture achieves all three benefits of Multus networking for Fusion Data Foundation deployments.

The following diagram shows the cluster network architecture for a Fusion Data Foundation deployment using the Multus public network:

Figure 1. Fusion Data Foundation deployment using Multus public network



This architecture achieves all the three benefits of Multus networking:

- Storage latency is reduced by avoiding the software-defined pod network
- Storage system gets dedicated bandwidth that avoids the noisy-neighbor interference
- Storage network is isolated from applications that prevent pods from snooping the storage network

However, if the public network has less throughput than the Pod network, storage traffic might not be faster than a traditional deployment.

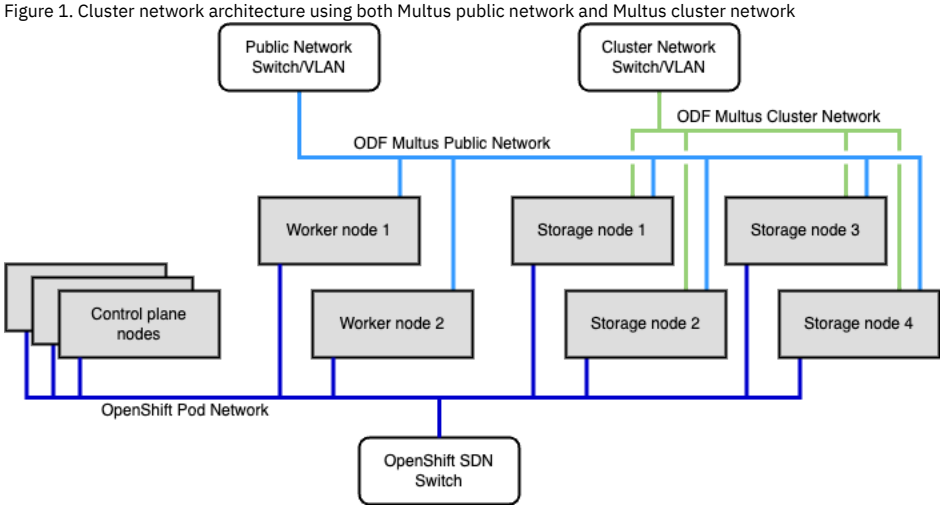
Public network is the recommended architecture for the majority of Fusion Data Foundation cluster deployments. In this architecture, all storage traffic uses the dedicated public network, which includes I/O traffic from applications, data replication traffic within the storage system, and data recovery traffic also within the storage system.

Multus public and cluster networks

Using both Multus public and cluster networks simultaneously provides dedicated networks for different types of storage traffic.

The following diagram shows the cluster network architecture for a Fusion Data Foundation deployment using both Multus public network and Multus cluster network:

Figure 1. Cluster network architecture using both Multus public network and Multus cluster network



When both public and cluster networks are used simultaneously, all three benefits of Multus are achieved. In this case, each storage network performs a specific role in the Fusion Data Foundation cluster. The public network handles only the I/O traffic from applications and the cluster network handles storage replication and recovery traffic.

The cluster network handles data replication traffic on a regular basis. Assuming Fusion Data Foundation's default 3x replication, the cluster network handles 2x the application I/O traffic for each data write to bring the total number of storage data copies up to 3x. During a storage node or zone failure event, the cluster network also handles recovering data in the cluster.

In this architecture, the size of the cluster network is recommended to be 2-3x larger than the public network. Larger sizing keeps the failover times short, but it also results in a large amount of bandwidth being wasted during normal cluster operations. Overall storage bandwidth during a storage failure event might be limited by either the public or cluster network depending on the size of either network.

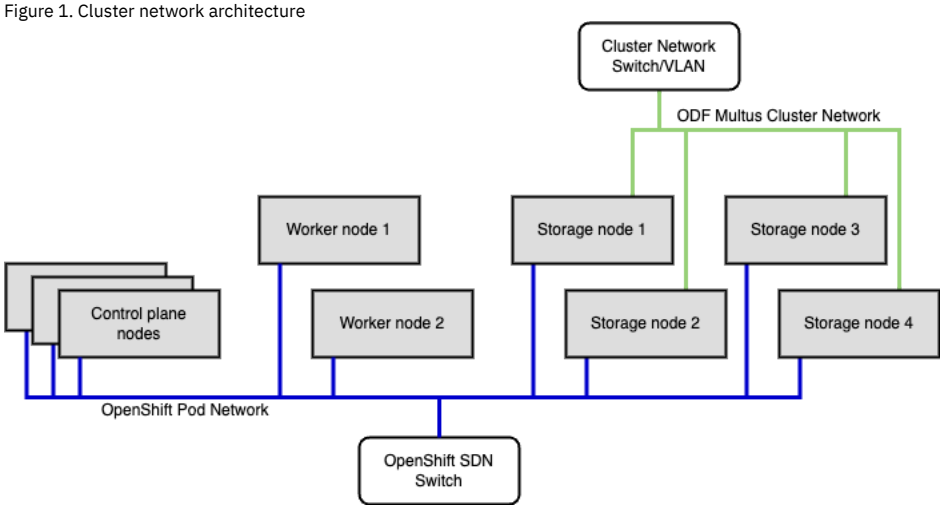
This architecture is more complicated than the recommended public-only architecture and has many more factors that could affect user SLOs. Hence, as an alternative, Fusion Data Foundation suggests bonding all storage network interfaces together to create a single, larger public network instead.

Multus cluster network

The cluster network architecture partially achieves Multus benefits by dedicating a network for replication and recovery traffic.

The following diagram shows the cluster network architecture for a Fusion Data Foundation deployment using Multus cluster network:

Figure 1. Cluster network architecture



In the cluster network architecture, Multus benefits are partially achieved. The cluster network takes the replication and recovery roles. The OpenShift Pod network takes the role of handling storage I/O from applications. This ensures noisy-neighbor interference is minimized, but application I/O can still be affected by other application networking, and vice versa.

Application I/O to the storage system is expected to have higher latency than other Multus architectures due to the software-defined Pod network. Similarly, the storage network is not isolated from applications. For example, a privileged Pod could still snoop storage traffic.

Multus networking requirements

Learn about the general, routing, and address space sizing requirements for implementing Multus networking in your deployment.

This section covers the following requirements for using Multus networking on a Fusion Data Foundation deployment.

- General requirements
- Routing requirements
- Address space sizing requirements

- **General requirements**
  The general requirements for Multus networking cover interface naming, connectivity, and hardware specifications.
- **Routing requirements**
  Routing requirements ensure proper connectivity between OpenShift nodes and Fusion Data Foundation storage through Multus networks.
- **Address space sizing**
  Address space sizing requirements ensure sufficient IP addresses for Fusion Data Foundation storage pods and OpenShift nodes.

General requirements

The general requirements for Multus networking cover interface naming, connectivity, and hardware specifications.

The general requirements for Multus networking are as follows:

- Interface used for the storage networks must have the same interface name on each OpenShift storage and worker node.
- Interfaces must all be connected to the same underlying network.
- When both public and cluster networks are used, they should not be connected to each other.
- At least one additional network for storage is required.
- Storage networks must be at least 10 Gbps. Bonded links are supported and suggested.

Fusion Data Foundation with Multus creates many virtual MAC addresses on each storage network host interface. This requires network cards and switch hardware to support promiscuous mode. Some smart switches have security processes that inspect each packet, and these switches can bog down significantly due to the high number of MAC addresses for Fusion Data Foundation. It is best to disable smart switch security features that require advanced processing.

Similarly, virtual or software switches might struggle to process the large number of MAC addresses. Some are not able to disable this functionality. Therefore, do not use virtual switches without a support exception.

Routing requirements

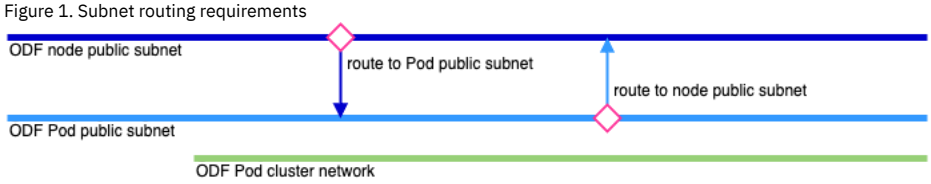
Routing requirements ensure proper connectivity between OpenShift nodes and Fusion Data Foundation storage through Multus networks.

For Fusion Data Foundation Persistent Volume Claims (PVCs) to work properly, the Ceph-CSI driver pods deployed to each OpenShift node (workers and storage) must use host networking. Ceph-CSI drivers must also be able to reach Fusion Data Foundation Ceph cluster through the Multus public network. This means that each OpenShift node must have a connection to the public network, and vice versa.

To make this as simple as possible without risk of IP overlap issues, select a different subnet (IP range) for node connections to the public network and Pod connections to the public network. Set up routing on the node public subnet and Pod public subnet such that they route to each other. The final result is that the node public subnet and Pod public subnet act together as a single contiguous public network.

If the cluster network is used, do not route the Pod public network or node public network to the cluster network. The cluster network should be independent and isolated. The following diagram illustrates subnet routing requirements:

Figure 1. Subnet routing requirements



Address space sizing

Address space sizing requirements ensure sufficient IP addresses for Fusion Data Foundation storage pods and OpenShift nodes.

Fusion Data Foundation with Multus requires an IP address for each Fusion Data Foundation Ceph storage daemon. Networks must have enough IP addresses to account for the number of storage pods that gets attached to the network, and some additional space to account for failover events. Expanding networks in the future might require downtime, so allocating space for the largest foreseeable cluster size is recommended.

For Multus public network, the following ranges must be planned:

- The Pod public network address space must include enough IPs for the total number of Fusion Data Foundation pods running in the openshift-storage namespace
- The node public network address space must include enough IPs for the total number of OpenShift nodes (worker + storage)

For Multus cluster network, the following range must be planned:

- The Pod cluster network address space must include enough IPs for the total number of OSD pods running in the openshift-storage namespace

Recommendations for Multus networking

This section describes the general and address range recommendations for Multus networking on Fusion Data Foundation deployment.

- **General recommendations for Multus networking**
  Red Hat Ceph networking recommendations provide useful guidance for Fusion Data Foundation Multus networking.
- **Address ranges**
  Recommended IP ranges for Multus networks ensure sufficient address space for Fusion Data Foundation deployments.

General recommendations for Multus networking

Red Hat Ceph networking recommendations provide useful guidance for Fusion Data Foundation Multus networking.

Red Hat Ceph underpins Fusion Data Foundation. Red Hat Ceph networking recommendations are useful for understanding more about Fusion Data Foundation Multus networking and for finding specific recommendations. For more information, see Network considerations for Red Hat Ceph Storage that also applies to Fusion Data Foundation networking.

Address ranges

Recommended IP ranges for Multus networks ensure sufficient address space for Fusion Data Foundation deployments.

The following table recommends IP ranges for each Multus network or subnetwork:

Table 1. Recommended IP ranges

Network	Network range CIDR	Approximate maximum	Public or cluster network
Pod public network	192.168.240.0/21	Total of 1600 Fusion Data Foundation pods	Public network
Pod cluster network	192.168.248.0/22	800 OSDs	Cluster network
Public node network	192.168.252.0/23	400 OpenShift worker+storage nodes	Public network

This should suffice for most organizations. The recommendation uses the last 6.25% (1/16th) of the reserved private address space (192.168.0.0/16). Approximate maximum Fusion Data Foundation and OpenShift sizing is noted, which accounts for 25% overhead to gracefully handle failure events.

Planning host interface connections

Plan network interface connections for OpenShift nodes based on node type and storage network requirements.

OpenShift nodes can be separated into following three types:

- OpenShift control plane nodes
- OpenShift worker nodes
- Fusion Data Foundation storage nodes

Each type requires different connections to the storage cluster as follows:

- All OpenShift nodes require access to the OpenShift machine network
- Control plane nodes do not require any connection to storage networks
  - If Fusion Data Foundation PVCs are needed on control plane nodes (not recommended), treat them as worker nodes
  - If control plane nodes are used as Fusion Data Foundation storage nodes, treat them as storage nodes
- Worker nodes run Ceph-CSI driver daemons to mount PVC storage to Pods and therefore require a connection to the Fusion Data Foundation public network (if used)
- Storage nodes require access to all used storage networks

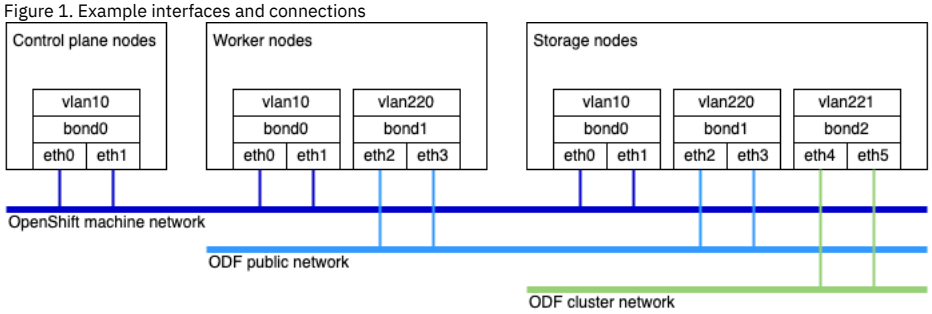
- **Example interfaces and connections**
  Example interfaces and connections illustrate how different node types can be configured for Multus networking.
- **Recommended host interface connection**
  The recommended host interface connection uses bonded interfaces for a public network-only architecture.

Example interfaces and connections

Example interfaces and connections illustrate how different node types can be configured for Multus networking.

The following diagram shows example interfaces and connections that might be planned for each node type:

Figure 1. Example interfaces and connections



This example includes cluster network for completeness.

In this example, **`vlan220`** (public) and **`vlan221`** (cluster) are the interfaces to use for Fusion Data Foundation configuration. If VLANs are unused in the deployed environment, **`bond1`** and **`bond2`** would be used instead. If no bonds or VLANs are used, host interfaces (for example, **`eth2`**, **`eth4`**) would be used instead. When using **`Whereabouts`** IPAM, IP addresses are not required on Multus parent interfaces such as NICs, bonds, or VLANs (for example, **`eth2`**–**`eth4`**, **`bond1`**–**`bond2`**, **`vlan220`**–**`vlan221`**).

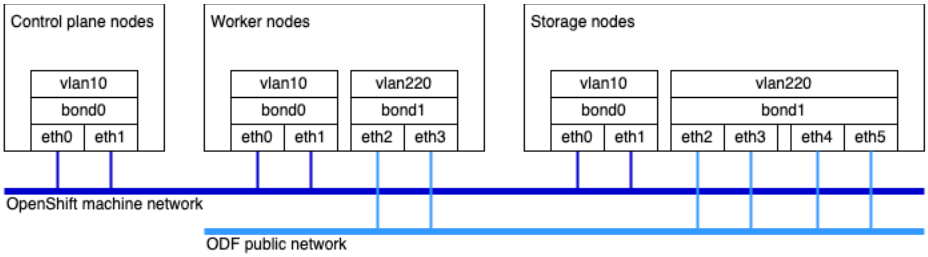
Important: The "ODF public network" in the diagram accounts for both the Fusion Data Foundation Pod public network and Fusion Data Foundation node public network. The OpenShift machine network is used as a backbone for the OpenShift Pod network and is included for reference only. Do not mistake this network for the ODF node public network. Do not plan or execute routing that involves the machine network.

Recommended host interface connection

The recommended host interface connection uses bonded interfaces for a public network-only architecture.

For a Fusion Data Foundation Multus architecture using only the public network (recommended for most users), the same nodes from the preceding example can be used. **`eth0`**, **`eth1`**, **`eth2`**, and **`eth4`** would be bonded as **`bond1`** and tagged as **`vlan220`**. This architecture ensures that storage nodes have additional available bandwidth for replication and recovery needed by Fusion Data Foundation Ceph OSDs. **`vlan220`** is be used for Fusion Data Foundation configuration. However, many variations of valid environments exist.

Figure 1. Recommended host interface connection



There is no requirement to dedicate extra public network bandwidth to storage nodes, though this is beneficial. **`eth4`** and **`eth5`** can be omitted from storage nodes entirely to dedicate the same bandwidth to both worker and storage nodes. This design has been commonly seen in customer environments.

Alternatively, **`eth2`** and **`eth3`** can be higher-throughput interfaces on storage nodes (for example, 10Gbps on worker nodes and 25Gbps on storage nodes). This allows for more storage node bandwidth while still using only two interfaces on storage nodes.

Multus network configuration

Configure Multus networks using **`NetworkAttachmentDefinition`** objects with **`macvlan`** CNI plugin and whereabouts IPAM plugin for IPv4 or IPv6 Pod network connections.

Multus networks are configured with **`NetworkAttachmentDefinition`** objects. These objects configure Pod network connections. OpenShift Network Attachment Definitions support many options, however, Fusion Data Foundation supports only a few that are outlined as follows.

- **`macvlan`** is the only supported CNI plugin
- **`whereabouts`** is the only supported IP Address Management (IPAM) plugin

Diverging from these selections requires a support exception.

Address range values must also match those selected for the cluster. The example cluster, example host interfaces, and recommended network ranges are used here. Modify them as needed in the deployed environment.

You can configure your Multus networks to use IPv4 or IPv6. Multus networks can only be configured to use either IPv4 or IPv6. Mixing modes is not supported.

- **Public NetworkAttachmentDefinition**
  The public network **`NetworkAttachmentDefinition`** configures Pod connections to the Multus public network.
- **Cluster NetworkAttachmentDefinition**
  The cluster network **`NetworkAttachmentDefinition`** configures Pod connections to the Multus cluster network.
- **Node configuration for Multus public network**
  Configure OpenShift worker and storage nodes to route host traffic to pods on the public network by using **`NodeNetworkConfigurationPolicy`** objects with the NMState Operator.

Public `NetworkAttachmentDefinition`

The public network **`NetworkAttachmentDefinition`** configures Pod connections to the Multus public network.

The recommended Pod public network configuration for the example is as follows. Omit this if the Multus public network is not used. Ensure to include the **`routes`** section that allows Pods on the public network to reach nodes through the public network.

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: public-net
  namespace: openshift-storage
spec:
  config: |
    {
      "cniVersion": "0.3.1",
      "type": "macvlan",
      "master": "vlan220", # host public network interface
      "mode": "bridge",
      "ipam": {
        "type": "whereabouts",
        "range": "192.168.240.0/21", # pod public network range
        "routes": [
          {"dst": "192.168.252.0/23"} # node public network range
        ]
      }
    }
```

Cluster `NetworkAttachmentDefinition`

The cluster network **`NetworkAttachmentDefinition`** configures Pod connections to the Multus cluster network.

The recommended Pod cluster network configuration for the example is as follows. Omit this if the Multus cluster network is not used (recommended for most users). This configuration should not have a **`routes`** section.

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: cluster-net
  namespace: openshift-storage
spec:
  config: |
    {
      "cniVersion": "0.3.1",
      "type": "macvlan",
      "master": "vlan221", # host cluster network interface
      "mode": "bridge",
      "ipam": {
        "type": "whereabouts",
        "range": "192.168.248.0/22" # pod cluster network range
      }
    }
```

Node configuration for Multus public network

Configure OpenShift worker and storage nodes to route host traffic to pods on the public network by using **`NodeNetworkConfigurationPolicy`** objects with the NMState Operator.

OpenShift worker and storage nodes must be configured to route host traffic to the Pods on the public network through the host public network interface.

The recommended way to configure nodes is using OpenShift **`NodeNetworkConfigurationPolicy`** objects. This method can be supported for all OpenShift users regardless of deployment strategy. This method requires the NMState Operator to be installed and enabled. For more information, see Kubernetes NMState Operator.

Each node must obtain an IP address on the ODF public network in the node public network address range. Static IP address management is the only IPAM method that can be supported for any OpenShift cluster. Thus, static management is Fusion Data Foundation supports only the static management method. This requires one **`NodeNetworkConfigurationPolicy`** object per host. The template that can be used to configure a host is shown in the following.

Important: The following template creates an interface called "shim" interface on each host. The shim interface uses the host public network interface (for example, **`vlan220`**) as a parent. The static IP is then given to the shim interface and not to the parent. Similarly, routing uses the shim. This is a critical detail. Macvlan disallows the virtual network of connected Pods on any given host from reaching the host directly or through switch hairpin. Without the shim interface, Fusion Data Foundation does not function properly. Do not attempt to set up the Fusion Data Foundation Multus public network without configuring the shim interface.

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: ceph-public-net-shim-{{NODE_NAME}}
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ""
    kubernetes.io/hostname: {{NODE_NAME}}
  desiredState:
    interfaces:
      - name: odf-pub-shim
        description: Shim interface to connect to ODF public network
        type: mac-vlan
        state: up
        mac-vlan:
          base-iface: vlan220 # host public network interface
          mode: bridge
          promiscuous: true
        ipv4:
          enabled: true
          dhcp: false
          address:
            - ip: 192.168.252.1 # static IP in node public network range
              prefix-length: 23 # node public network range mask
    routes:
      config:
        - destination: 192.168.240.0/21 # pod public network range
          next-hop-interface: odf-pub-shim
```

First, follow comments in the template to update the base template for the environment being deployed. Then, for each node, copy the template, and fill in **`{{NODE_NAME}}`** and a unique static IP for each node.

Preinstallation checklist for Multus configuration

Complete these preinstallation tasks before you deploy Multus networking for your storage cluster.

After all the previous sections have been planned or executed, Fusion Data Foundation preinstallation can begin. The following checklist of preinstallation items should be completed before proceeding further.

1. Read the Multus architecture document in its entirety.
2. Select which Fusion Data Foundation Multus networks to use.
3. Plan host interface connections.
4. Deploy OpenShift.
5. Select network address ranges to use.
6. Write **`NetworkAttachmentDefinition`** manifests for Pods.
7. Write **`NodeNetworkConfigurationPolicy`** manifests for nodes.
8. Deploy the manifests to the OpenShift cluster.

9. Install Fusion Data Foundation, but do not deploy a StorageCluster yet.
10. Run the Multus validation tool to check preinstallation deployments (covered in the next section).

Validating and debugging Multus configuration

Fusion Data Foundation has a tool to perform basic connectivity tests prior to Fusion Data Foundation StorageCluster deployment. This Multus validation tool is intended to be a self-help tool that returns a list of details that might be incorrect based on the validation steps that pass or fail.

This tool performs basic connection and IP availability checking. It cannot detect every possible misconfiguration, and it cannot detect network performance issues. Ensure due diligence for each cluster deployment.

For more information, see Multus prerequisites validation tool.

When encountering an issue with the tool, check the tool output for suggestions. Use events associated with Pods that the tool runs (for example, using `oc -n openshift-storage describe pods`) to debug further. If issues persist despite debugging, collect Fusion Data Foundation must-gather, and open a Fusion Data Foundation help ticket that includes full tool output and the must-gather tarball. For more information, see Downloading log files and diagnostic information using must-gather.

For best results, use a configuration file with the Multus validation tool. The following configuration is recommended, which can be updated by following the comments as needed.

```
publicNetwork: "public-net" # comment out if public network is unused
# clusterNetwork: "cluster-net" # uncomment if used


# for airgapped environments, mirror this image, then update this config to reference the mirror
nginxImage: "quay.io/nginx/nginx-unprivileged:stable-alpine"

namespace: "openshift-storage"
serviceAccountName: "rook-ceph-system"

nodeTypes:
  storage-nodes:
    osdsPerNode: 12 # set to the max number of OSDs per storage node
    otherDaemonsPerNode: 16
    placement:
      nodeSelector:
        cluster.ocs.openshift.io/openshift-storage: ""
      tolerations:
        - {"key":"node.ocs.openshift.io/storage","value":"true"}
  worker-nodes:
    osdsPerNode: 0
    otherDaemonsPerNode: 6
    placement:
      nodeSelector:
      tolerations:
```

If this recommendation proves to be inadequate or out of date, use the tool to generate a new configuration using `./rook multus validation config dedicated-storage-nodes`.

Fusion Data Foundation installation after validating Multus configuration

After you validate connectivity by using the Multus validation tool, you can deploy the Fusion Data Foundation storage cluster and select the appropriate Multus networks during deployment.

- **Post-installation checks**
  After you deploy Fusion Data Foundation with Multus, verify that network assignments, storage access, and cluster behavior work as expected.

Post-installation checks

After you deploy Fusion Data Foundation with Multus, verify that network assignments, storage access, and cluster behavior work as expected.

In addition to the Fusion Data Foundation health checking, ensure that Multus is configured as expected by performing the following:

- Using the ODF toolbox or `odf-cli` tool, run `ceph osd dump` to verify that OSDs have an IP in the cluster network.
  In the output, ensure that OSD daemons have an IP in the public network (if used). OSDs should additionally have an IP in the cluster network, if the Multus cluster network was used. If it was left empty, OSDs should only list public network IPs. If a filesystem has been deployed, ensure that MDS daemons have an IP in the public network (if used).
- Create a test application Deployment in a new namespace that uses Fusion Data Foundation PVCs.
  Ensure that the test application Pod can write to the mounted PVC volumes. Delete the Deployment's Pod, then wait for it to reschedule to a new node. Ensure that the previous data can be read back, and ensure new data can be written.
- Run some test load generators in the cluster to write to Fusion Data Foundation PVCs. After the Fusion Data Foundation cluster has filled to 20%, induce a hard failure of a node or failure zone. Leave the node or zone in failure state for an hour, and observe the Fusion Data Foundation cluster status and Fusion Data Foundation Pod statuses. Cluster health can be expected to show **`HEALTH_WARN`**, but it should not show **`HEALTH_ERR`**. One or two Fusion Data Foundation Pod failures may be okay, but several failures could indicate a problem.

Enabling Multus support on an existing Fusion Data Foundation cluster

This procedure describes how to configure Multus networking for an existing Fusion Data Foundation deployment.

Before you begin

- Before proceeding, review the Multus requirements and architecture described in Multus architecture for Fusion Data Foundation.
- Ensure that the environment meets these prerequisites before migrating the cluster from a non-Multus network to a Multus-enabled network.

Procedure

1. Configure the nodes.
   As described in the architecture documentation, the recommended method for configuring node-level network settings is to use the **`NodeNetworkConfigurationPolicy`** (NNCP).

   Using NNCP requires the NMState Operator to be installed and enabled on the cluster. NNCPs are required only when a **public network** is used.

   Example NodeNetworkConfigurationPolicy for node **`compute-0`**:

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  annotations:
    nmstate.io/webhook-mutating-timestamp: "1743035283553699179"
  name: ceph-public-net-shim-compute-0
spec:
  desiredState:
    interfaces:
    - description: Shim interface used to connect host to Fusion Data Foundation public Multus network
      ipv4:
        address:
        - ip: 192.168.252.1
          prefix-length: 24
        dhcp: false
        enabled: true
      mac-vlan:
        base-iface: ens224
        mode: bridge
        promiscuous: true
      name: odf-pub-shim
      state: up
      type: mac-vlan
  nodeSelector:
    kubernetes.io/hostname: compute-0
    node-role.kubernetes.io/worker: ""
```

2. Create NetworkAttachmentDefinition (NAD) objects.
   Before creating NADs, review the NAD requirements and the NAD configuration guidance provided in the Multus documentation.

   The following examples show both public and cluster network NADs. Only the NADs required for your deployment need to be created.

   Example public network NAD:

```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  name: public-net
  namespace: default
spec:
  config: '{"cniVersion": "0.3.1", "type": "macvlan", "master": "ens224", "mode":
    "bridge", "ipam": {"type": "whereabouts", "range": "192.168.20.0/24", "routes":
    [{"dst": "192.168.252.0/24"}]}}'
```

   Example cluster network NAD:

```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  name: cluster-net
  namespace: default
spec:
  config: '{"cniVersion": "0.3.1", "type": "macvlan", "master": "ens224", "mode":
    "bridge", "ipam": {"type": "whereabouts", "range": "192.168.30.0/24"}}'
```

3. Run the Multus validation tool.
   Run the validation tool as described in the Multus validation documentation. A successful validation produces output similar to the following:

```
RESULT: multus validation test succeeded
```

   This result confirms that the network configuration is valid.

4. Edit the StorageCluster custom resource.
   After the validation tool succeeds, update the **`StorageCluster`** CR to configure Multus networking. Modify the **`network`** section to include the required providers and selectors.

   Example StorageCluster networking configuration:

```
network:
  multiClusterService: {}
  provider: multus
  selectors:
    cluster: default/cluster-net
    public: default/public-net
```

This example includes both cluster and public networks. Specify only the networks required for your deployment.

5. Move MDS pod to Multus network manually as it does not get moved automatically.
   Restart the MDS pods after the OSD pods are moved to Multus network by running the following command:

```
num_osds="$(oc --namespace openshift-storage get pod -l app=rook-ceph-osd --no-headers | wc -l)"

while true; do
  osds_with_multus="$(oc --namespace openshift-storage get pod -l app=rook-ceph-osd | grep -c
'k8s.v1.cni.cncf.io/networks')"
  [[ $osds_with_multus -eq $num_osds ]] &amp;&amp; break
  sleep 10
done

while true; do
  osds_running="$(oc --namespace openshift-storage get pod -l app=rook-ceph-osd | grep -c 'Running')"
  [[ $osds_running -eq $num_osds ]] &amp;&amp; break
  sleep 10
done

oc --namespace openshift-storage delete pod -l app=rook-ceph-mds
```

   This gets the OSD pod count. Then it checks OSD pods with Multus network and confirms that the OSD pods are running. Later it deletes the MDS pods.

6. Validate the Ceph network configuration.
   After updating the StorageCluster, verify that Ceph OSDs are using the Multus network addresses.

   Run the following command:

```
$ ceph osd dump | grep osd
```

   Example output:

```
osd.0 ... [v2:192.168.20.8:6800/... ,v1:192.168.20.8:6801/...] [v2:192.168.30.1:6800/...,v1:192.168.30.1:6801/...]
osd.1 ... [v2:192.168.20.9:6800/... ,v1:192.168.20.9:6801/...] [v2:192.168.30.2:6800/...,v1:192.168.30.2:6801/...]
osd.2 ... [v2:192.168.20.10:6800/... ,v1:192.168.20.10:6801/...] [v2:192.168.30.3:6800/...,v1:192.168.30.3:6801/...]
```

   In this example:

   - The public network range is: `192.168.20.0/24`
   - The cluster network range is: `192.168.30.0/24`

   The OSDs show addresses in both ranges, confirming successful Multus integration.

Multus prerequisites validation tool

The Multus prerequisites validation tool validates OpenShift configuration, **`NetworkAttachmentDefinitions`**, and network compatibility before Fusion Data Foundation installation.

This is an interactive tool to help support in-the-field debug and resolution of common configuration issues that affect Multus clusters. It runs a validation test that determines whether the current **`NetworkAttachmentDefinition`**, and system configurations support Fusion Data Foundation with Multus.

It is a long-running test. It starts up a web server and many clients to verify that Multus network communication works properly.

It does not perform any load testing. Networks that cannot support high volumes of Ceph traffic might still encounter runtime issues. This might be noticeable with high I/O load or during OSD rebalancing (for example, during node/disk failure or during Fusion Data Foundation upgrade). Therefore, it is recommended to perform network load testing to ensure that base network configurations meet user requirements. To know more about OSD rebalancing, see Rebalancing in *Ceph documentation*.

- **How to use validation tool**
  Run the validation tool from an OpenShift administrator shell after configuring the Multus **`NetworkAttachmentDefinitions`** and installing the Fusion Data Foundation operator, but before installing a Fusion Data Foundation StorageCluster.
- **Troubleshooting validation tool**
  If the test fails when you run the validation tool, it suggests some things to check based on the failure condition. This helps to resolve common configuration issues quickly.
- **How to get help**
  How to get help when cannot resolve the issue.

How to use validation tool

Run the validation tool from an OpenShift administrator shell after configuring the Multus **`NetworkAttachmentDefinitions`** and installing the Fusion Data Foundation operator, but before installing a Fusion Data Foundation StorageCluster.

Procedure

1. Access the **`rook-ceph-operator`** pod.
   For example:

```
oc rsh <rook-ceph-operator-pod-name>
```

2. Run the tool.
   The tool provides extensive help text. When you run the tool, you get the most up-to-date version of the tool.

   For example:

   ```
   ./rook multus validation run -h
   ```

What to do next

The validation tool supports a configuration file that allows configuring the number of test daemons on different types of nodes. You can use configuration file to test CSI+Ceph+OSD placement on storage nodes while testing only CSI placement on non-storage nodes. The tool includes configuration file examples with commented documentation to help you get started quickly.

```
./rook multus validation config -h
```

Troubleshooting validation tool

If the test fails when you run the validation tool, it suggests some things to check based on the failure condition. This helps to resolve common configuration issues quickly.

- For diagnosing issues with the tool itself, use the --log-level DEBUG for getting additional details.
- If pods are not starting, you can expect the following issues:
  - Problem with NAD configuration.
  - The `oc pod describe` command contains the error as an event.
  - Some NICs do not support enough virtual ports or VLANs for the additional MAC addresses.
- If pods are starting but not communicating, you can expect the following issues:
  - Network design or configuration might contain errors
  - Network switch might be blocking sub-interface MAC addresses or IPs (check promiscuous mode settings)
  - Switch firewall
  - System or NAD Linux networking configurations might be blocking the traffic
  - SOS report has good information to look at next
    - `ip_netns_exec_*_address_*` and `ip_netns_exec_*_route_*` from container namespaces

How to get help

How to get help when cannot resolve the issue.

If you are having trouble resolving the issue based on these troubleshooting steps described in Troubleshooting validation tool, a substantial amount of information should be collected for getting help.

The tool lists resources that should be collected into an archive file in this case. For OpenShift and Fusion Data Foundation, the following OpenShift tools contains the information needed for further debugging:

- Network diagrams
- ODF must-gather
- OCP must-gather
- OCP must-gather network logs
- SOS reports from all nodes on which test pods are running (https://access.redhat.com/solutions/3530881)

Multus is highly integrated among various product layers, and issues are most likely to be related to configuration or the physical hardware environment rather than product bugs. Initial investigation at the engineering layer involves ODF, OpenShift, and possibly RHEL networking experts to rule out configuration or environment.

- General Multus bugs/help: OCP Jira: https://issues.redhat.com/projects/OCPBUGSM
- FDF Multus bugs: ODF jira: https://issues.redhat.com/projects/DFBUGS
- Contact IBM Support

Planning Fusion Data Foundation deployment

Use this information for important considerations when planning your IBM Fusion Data Foundation deployment.

- **Storage cluster deployment approaches**
  The growing list of operating modalities is an evidence that flexibility is a core tenet of IBM Fusion Data Foundation. Use this information to help you to select the most appropriate approach for your environments.
- **Node types**
  Nodes run the container runtime, as well as services, to ensure that the containers are running, and maintain network communication and separation between the pods.
- **Internal storage services**
  Internal storage services are available for use with Red Hat OpenShift Container Platform with various platforms.
- **Provider storage services**
  You can deploy Fusion Data Foundation in Provider mode to make it available for consumption to the Red Hat OpenShift Container Platform that runs on the bare

metal infrastructure:

- **External storage services**
  External storage services are available for use with IBM Fusion Data Foundation.
- **Security considerations**
  Understand these security considerations when planning for IBM Fusion Data Foundation.
- **Subscription offerings**
  IBM Fusion Data Foundation subscription is based on “core-pairs,” similar to Red Hat OpenShift Container Platform. The IBM Fusion Data Foundation 2-core subscription is based on the number of logical cores on the CPUs in the system where OpenShift Container Platform runs.
- **Infrastructure requirements**
  Understand the infrastructure requirements when planning for Fusion Data Foundation.
- **Network requirements**
  Use this section to understand the different network considerations when planning deployments.
- **Disaster Recovery**
  Disaster Recovery (DR) helps an organization to recover and resume business critical functions or normal operations when there are disruptions or disasters.
- **Supported and unsupported features for IBM Power and IBM Z infrastructures**
  Use this information to understand which features are supported and unsupported in IBM Power and IBM Z infrastructures.

Storage cluster deployment approaches

The growing list of operating modalities is an evidence that flexibility is a core tenet of IBM Fusion Data Foundation. Use this information to help you to select the most appropriate approach for your environments.

The main difference between internal and external modes is that you can deploy IBM Fusion Data Foundation either entirely within Red Hat OpenShift Container Platform (Internal approach) or to make available the services from a cluster running outside of Red Hat OpenShift Container Platform (External approach).

Internal mode

Deployment of IBM Fusion Data Foundation entirely within Red Hat OpenShift Container Platform has all the benefits of operator based deployment and management. You can use the internal-attached device approach in the graphical user interface (GUI) to deploy Fusion Data Foundation in internal mode using the local storage operator and local storage devices.

Ease of deployment and management are the highlights of running Fusion Data Foundation services internally on OpenShift Container Platform. There are two different deployment modalities available when Fusion Data Foundation is running entirely within Red Hat OpenShift Container Platform:

- Simple
- Optimized

Simple deployment
: Fusion Data Foundation services run co-resident with applications. The operators in Red Hat OpenShift Container Platform manages these applications.
A simple deployment is best for situations where,

  - Storage requirements are not clear.
  - Fusion Data Foundation services runs co-resident with the applications.
  - Creating a node instance of a specific size is difficult, for example, on bare metal.

  For Fusion Data Foundation to run co-resident with the applications, the applications must have local storage devices, or portable storage devices attached to them dynamically, like EBS volumes on EC2, or vSphere Virtual Volumes on VMware, or SAN volumes.
  Note: PowerVC dynamically provisions the SAN volumes.

Optimized deployment
: IBM Fusion Data Foundation services run on dedicated infrastructure nodes. Red Hat OpenShift Container Platform manages these infrastructure nodes.
An optimized approach is best for situations when,

  - Storage requirements are clear.
  - IBM Fusion Data Foundation services run on dedicated infrastructure nodes.
  - Creating a node instance of a specific size is easy, for example, on cloud, virtualized environment, and so on.

External mode

IBM Fusion Data Foundation exposes the IBM Storage Ceph services running outside of the OpenShift Container Platform cluster as storage classes.

The external approach is best used in the following use cases:

- Storage requirements are significant (600+ storage devices).
- Multiple OpenShift Container Platform clusters need to consume storage services from a common external cluster.
- Another team, Site Reliability Engineering (SRE), storage, and so on, need to manage the external cluster providing storage services. Possibly a preexisting one.

- **Internal approach**
  Deployment of IBM Fusion Data Foundation entirely within Red Hat OpenShift Container Platform has all the benefits of operator based deployment and management. You can use the internal-attached device approach in the graphical user interface (GUI) to deploy Fusion Data Foundation in internal mode using the local storage operator and local storage devices.
- **Provider mode approach**
  Fusion Data Foundation can be configured to run in Provider mode and Fusion Data Foundation client can connect to it and create storage classes that can be used by workloads.
- **External mode**
  IBM Fusion Data Foundation exposes the IBM Storage Ceph services running outside of the OpenShift Container Platform cluster as storage classes.

Internal approach

Deployment of IBM Fusion Data Foundation entirely within Red Hat OpenShift Container Platform has all the benefits of operator based deployment and management. You can use the internal-attached device approach in the graphical user interface (GUI) to deploy Fusion Data Foundation in internal mode using the local storage operator and local storage devices.

Ease of deployment and management are the highlights of running Fusion Data Foundation services internally on OpenShift Container Platform. There are two different deployment modalities available when Fusion Data Foundation is running entirely within Red Hat OpenShift Container Platform:

- Simple
- Optimized

Simple deployment
: Fusion Data Foundation services run co-resident with applications. The operators in Red Hat OpenShift Container Platform manages these applications.
A simple deployment is best for situations where,

  - Storage requirements are not clear.
  - Fusion Data Foundation services runs co-resident with the applications.
  - Creating a node instance of a specific size is difficult, for example, on bare metal.

  For Fusion Data Foundation to run co-resident with the applications, the applications must have local storage devices, or portable storage devices attached to them dynamically, like EBS volumes on EC2, or vSphere Virtual Volumes on VMware, or SAN volumes.
  Note: PowerVC dynamically provisions the SAN volumes.

Optimized deployment
: IBM Fusion Data Foundation services run on dedicated infrastructure nodes. Red Hat OpenShift Container Platform manages these infrastructure nodes.
An optimized approach is best for situations when,

  - Storage requirements are clear.
  - IBM Fusion Data Foundation services run on dedicated infrastructure nodes.
  - Creating a node instance of a specific size is easy, for example, on cloud, virtualized environment, and so on.

Provider mode approach

Fusion Data Foundation can be configured to run in Provider mode and Fusion Data Foundation client can connect to it and create storage classes that can be used by workloads.

Provider mode approach can be effectively used in one of the following scenarios:

- When the storage consumption follows a hub and spoke paradigm where a Fusion Data Foundation provider acts as a hub providing multi-tenancy and multiple Fusion Data Foundation client clusters connect to it as spokes.
- When isolation of shared file and block storage per client cluster is required.
- When management of storage and applications is done by separate teams. For example, when a storage administrator manages the Fusion Data Foundation provider cluster and an application administrator manages the Fusion Data Foundation client cluster.

External mode

IBM Fusion Data Foundation exposes the IBM Storage Ceph services running outside of the OpenShift Container Platform cluster as storage classes.

The external mode is best used in the following use cases:

- Storage requirements are significant (600+ storage devices).
- Multiple OpenShift Container Platform clusters need to consume storage services from a common external cluster.
- Another team, Site Reliability Engineering (SRE), storage, and so on, need to manage the external cluster providing storage services. Possibly a preexisting one.

Node types

Nodes run the container runtime, as well as services, to ensure that the containers are running, and maintain network communication and separation between the pods.

In Fusion Data Foundation, there are three types of nodes, as described in Table 1.

Table 1. Types of nodes

Node Type	Description
Master	These nodes run processes that expose the Kubernetes API, watch and schedule newly created pods, maintain node health and quantity, and control interaction with underlying cloud providers.
Infrastructure (Infra)	Infra nodes run cluster level infrastructure services such as logging, metrics, registry, and routing. These are optional in OpenShift Container Platform clusters. To separate Fusion Data Foundation layer workload from applications, ensure that you use infra nodes for Fusion Data Foundation in virtualized and cloud environments. To create Infra nodes, you can provision new nodes labeled as `infra`. For more information, see How to use dedicated worker nodes for IBM Fusion Data Foundation

Node Type	Description
Worker	Worker nodes are also known as application nodes since they run applications. When Fusion Data Foundation is deployed in internal mode, you require a minimal cluster of 3 worker nodes. Ensure that the nodes are spread across 3 different racks, or availability zones, to ensure availability. For Fusion Data Foundation to run on worker nodes, you need to attach the local storage devices, or portable storage devices to the worker nodes dynamically. Fusion Data Foundation When IBM Fusion Data Foundation is deployed in external mode, it runs on multiple nodes. This allows Kubernetes to reschedule on the available nodes in case of a failure.

Note: Fusion Data Foundation requires the same number of subscriptions as OpenShift Container Platform. However, if Fusion Data Foundation is running on infra nodes, OpenShift does not require OpenShift Container Platform subscription for these nodes. Therefore, the Fusion Data Foundation control plane does not require additional OpenShift Container Platform and Fusion Data Foundation subscriptions. For more information, see Subscription Offerings.

Internal storage services

Internal storage services are available for use with Red Hat OpenShift Container Platform with various platforms.

IBM Fusion Data Foundation service is available for consumption internally to the Red Hat OpenShift Container Platform that runs on the following infrastructure:

- Amazon Web Services (AWS)
- Bare metal
- VMware vSphere
- Microsoft Azure
- Google Cloud
- Red Hat OpenStack 13 or higher (installer-provisioned infrastructure) [Technology Preview]
- IBM Power
- IBM Z and IBM LinuxONE

Creation of an internal cluster resource results in the internal provisioning of the Fusion Data Foundation base services, and makes additional storage classes available to the applications.

Provider storage services

You can deploy Fusion Data Foundation in Provider mode to make it available for consumption to the Red Hat OpenShift Container Platform that runs on the bare metal infrastructure:

The following can connect to the Fusion Data Foundation provider cluster:

- Clients Running on another bare metal cluster
- Hosted Cluster with bare metal
- Red Hat OpenShift Virtualization platform

External storage services

External storage services are available for use with IBM Fusion Data Foundation.

IBM Fusion Data Foundation can use IBM FlashSystem systems or make services from an external IBM Storage Ceph cluster available for consumption through OpenShift Container Platform clusters running on the following platforms:

- VMware vSphere
- Bare metal
- Red Hat OpenStack platform (Technology Preview)
- IBM Power
- IBM Z infrastructure

The Fusion Data Foundation operators create and manage services to satisfy Persistent Volume (PV) and Object Bucket Claims (OBCs) against the external services. External cluster can serve block, file and object storage classes for applications that run on OpenShift Container Platform. The operators do not deploy or manage the external clusters.

Security considerations

Understand these security considerations when planning for IBM Fusion Data Foundation.

- **FIPS-140-2**
  The Federal Information Processing Standard Publication 140-2 (FIPS-140-2) is a standard that defines a set of security requirements for the use of cryptographic modules. Law mandates this standard for the US government agencies and contractors and is also referenced in other international and industry specific standards.
- **Proxy environment**
  A proxy environment is a production environment that denies direct access to the internet and provides an available HTTP or HTTPS proxy instead.
- **Data encryption options**
  Encryption lets you encode your data to make it impossible to read without the required encryption keys. This mechanism protects the confidentiality of your data in the event of a physical security breach that results in a physical media to escape your custody. The per-PV encryption also provides access protection from other namespaces inside the same OpenShift Container Platform cluster. Data is encrypted when it is written to the disk, and decrypted when it is read from the disk. Working with encrypted data might incur a small penalty to performance.
- **Encryption in Transit**
  Enable IPsec so that all the network traffic between the nodes on the OVN-Kubernetes Container Network Interface (CNI) cluster network travels through an encrypted tunnel.

FIPS-140-2

The Federal Information Processing Standard Publication 140-2 (FIPS-140-2) is a standard that defines a set of security requirements for the use of cryptographic modules. Law mandates this standard for the US government agencies and contractors and is also referenced in other international and industry specific standards.

IBM Fusion Data Foundation uses the FIPS validated cryptographic modules. Red Hat Enterprise Linux OS/CoreOS (RHCOS) delivers these modules.

Currently, the Cryptographic Module Validation Program (CMVP) processes the cryptography modules. You can see the state of these modules in the Modules in Process List. For more up-to-date information, see the RHEL core crypto components Knowledgebase article on the Red Hat Customer Portal.
Note: Enable the FIPS mode on the OpenShift Container Platform, before you install Fusion Data Foundation. OpenShift Container Platform must run on the RHCOS nodes, as the feature does not support Fusion Data Foundation deployment on Red Hat Enterprise Linux 7 (RHEL 7).
For more information, see Install > Installation overview > Support for FIPS cryptography in Red Hat OpenShift Container Platform product documentation.

Proxy environment

A proxy environment is a production environment that denies direct access to the internet and provides an available HTTP or HTTPS proxy instead.

Red Hat OpenShift Container Platform is configured to use a proxy by modifying the proxy object for existing clusters or by configuring the proxy settings in the install-config.yaml file for new clusters.

IBM supports deployment of Fusion Data Foundation in proxy environments when OpenShift Container Platform has been configured. For more information, see Networking > Configuring network settings > Configuring the cluster-wide proxy in Red Hat OpenShift Container Platform product documentation.

Data encryption options

Encryption lets you encode your data to make it impossible to read without the required encryption keys. This mechanism protects the confidentiality of your data in the event of a physical security breach that results in a physical media to escape your custody. The per-PV encryption also provides access protection from other namespaces inside the same OpenShift Container Platform cluster. Data is encrypted when it is written to the disk, and decrypted when it is read from the disk. Working with encrypted data might incur a small penalty to performance.

Encryption is only supported for new clusters deployed using IBM Fusion Data Foundation 4.12 or higher. An existing encrypted cluster that is not using an external Key Management System (KMS) cannot be migrated to use an external KMS.

Previously, HashiCorp Vault was the only supported KMS for Cluster-wide and Persistent Volume encryption. Fusion Data Foundation now supports HashiCorp Vault KV secret engine API, versions 1 and 2.
Important:

- KMS is required for Storage Class encryption, and is optional for cluster-wide encryption.
- To start with, Storage class encryption requires a valid IBM Fusion Data Foundation subscription.

IBM works with the technology partners to provide this documentation as a service to the customers. However, IBM does not provide support for the Hashicorp product. For technical assistance with this product, contact Hashicorp.

- **Cluster-wide encryption**
  IBM Fusion Data Foundation supports cluster-wide encryption (encryption-at-rest) for all the disks and Multicloud Object Gateway operations in the storage cluster. Fusion Data Foundation uses Linux Unified Key System (LUKS) version 2 based encryption with a key size of 512 bits and the `aes-xts-plain64` cipher where each device has a different encryption key. The keys are stored using a Kubernetes secret or an external KMS. Both methods are mutually exclusive and you cannot migrate between methods.
- **Storage class encryption**
  You can encrypt persistent volumes (block only) with storage class encryption using an external Key Management System (KMS) to store device encryption keys.
- **CipherTrust manager**
  CipherTrust Manager is a Key Management System (KMS) provider supported for your deployment that uses the KMIP protocol to provide centralized key lifecycle management and is enabled during deployment.
- **Data encryption in-transit using messenger version 2 protocol of IBM Storage Ceph**
  Starting with Fusion Data Foundation version 4.14, you can use the messenger version 2 protocol of IBM Storage Ceph to encrypt data in-transit.

Cluster-wide encryption

IBM Fusion Data Foundation supports cluster-wide encryption (encryption-at-rest) for all the disks and Multicloud Object Gateway operations in the storage cluster. Fusion Data Foundation uses Linux Unified Key System (LUKS) version 2 based encryption with a key size of 512 bits and the **`aes-xts-plain64`** cipher where each device has a different encryption key. The keys are stored using a Kubernetes secret or an external KMS. Both methods are mutually exclusive and you cannot migrate between methods.

Encryption is disabled by default for block and file storage. You can enable encryption for the cluster at the time of deployment. The Multiloud Object Gateway supports encryption by default. See the deployment guides for more information.

Fusion Data Foundation supports cluster wide encryption with and without Key Management System (KMS). Cluster wide encryption with KMS is supported using the following service providers:

- HashiCorp Vault
- Thales Cipher Trust Manager

Security common practices require periodic encryption key rotation. IBM Fusion Data Foundation automatically rotates encryption keys stored in kubernetes secret (non-KMS) and Vault on a weekly basis. However, key rotation for Vault KMS must be enabled after the storage cluster creation and does not happen by default.

Note: Cluster wide encryption requires a valid Fusion Data Foundation advanced subscription.
Cluster wide encryption with HashiCorp Vault KMS provides two authentication methods:

- **Token**: This method allows authentication using vault tokens. A kubernetes secret containing the vault token is created in the **`openshift-storage`** namespace and is used for authentication. If this authentication method is selected then the administrator has to provide the vault token that provides access to the backend path in Vault, where the encryption keys are stored.
- **Kubernetes**: This method allows authentication with vault using serviceaccounts. If this authentication method is selected then the administrator has to provide the name of the role configured in Vault that provides access to the backend path, where the encryption keys are stored. The value of this role is then added to the **`ocs-kms-connection-details`** config map.

Note: Fusion Data Foundation on IBM Cloud platform supports Hyper Protect Crypto Services (HPCS) Key Management Services (KMS) as the encryption solution in addition to HashiCorp Vault KMS.
Important: IBM works with the technology partners to provide this documentation as a service to the customers. However, IBM does not provide support for the Hashicorp product. For technical assistance with this product, contact Hashicorp.

Storage class encryption

You can encrypt persistent volumes (block only) with storage class encryption using an external Key Management System (KMS) to store device encryption keys.

Persistent volume encryption is only available for RADOS Block Device (RBD) persistent volumes. See how to create a storage class with persistent volume encryption.

Storage class encryption is supported in Fusion Data Foundation with HashiCorp Vault KMS .

Note: Storage class encryption requires a validFusion Data Foundation advanced subscription.

CipherTrust manager

CipherTrust Manager is a Key Management System (KMS) provider supported for your deployment that uses the KMIP protocol to provide centralized key lifecycle management and is enabled during deployment.

IBM Fusion Data Foundation version 4.12 introduced Thales CipherTrust Manager as an additional Key Management System (KMS) provider for your deployment. Thales CipherTrust Manager provides centralized key lifecycle management. CipherTrust Manager supports Key Management Interoperability Protocol (KMIP), which enables communication between key management systems.

CipherTrust Manager is enabled during deployment.

Data encryption in-transit using messenger version 2 protocol of IBM Storage Ceph

Starting with Fusion Data Foundation version 4.14, you can use the messenger version 2 protocol of IBM Storage Ceph to encrypt data in-transit.

This provides an important security requirement for your infrastructure. You can enable in-transit encryption during deployment.

Encryption in Transit

Enable IPsec so that all the network traffic between the nodes on the OVN-Kubernetes Container Network Interface (CNI) cluster network travels through an encrypted tunnel.

By default, IPsec is disabled. You can enable it either during or after installing the cluster. If you need to enable IPsec after cluster installation, you must first resize your cluster MTU to account for the overhead of the IPsec ESP IP header.

For more information on how to configure the IPsec encryption, see Networking > Network security > Configuring IPsec encryption in Red Hat OpenShift Container Platform product documentation.

Subscription offerings

IBM Fusion Data Foundation subscription is based on “core-pairs,” similar to Red Hat OpenShift Container Platform. The IBM Fusion Data Foundation 2-core subscription is based on the number of logical cores on the CPUs in the system where OpenShift Container Platform runs.

As with OpenShift Container Platform:

- Fusion Data Foundation subscriptions are stackable to cover larger hosts.
- Cores can be distributed across as many virtual machines (VMs) as needed. For example, ten 2-core subscriptions provide 20 cores and in case of IBM Power a 2-core subscription at SMT level of 8 provides 2 cores or 16 vCPUs that can be used across any number of VMs.
- Fusion Data Foundation subscriptions are available with Premium or Standard support.

- **Cores versus vCPUs and hyperthreading**
  Making a determination about whether or not a particular system consumes one or more cores is currently dependent on whether or not that system has hyperthreading available. Hyperthreading is only a feature of Intel CPUs.
- **Splitting cores**
  Systems that require an odd number of cores need to consume a full 2-core subscription. For example, a system that is calculated to require only 1 core ends up consuming a full 2-core subscription once it is registered and subscribed.
- **Subscription requirements**
  IBM Fusion Data Foundation components can run on either OpenShift Container Platform worker or infrastructure nodes, for which you can use either Red Hat CoreOS (RHCOS) or Red Hat Enterprise Linux (RHEL) 8.4 as the host operating system. RHEL 7 is now deprecated. Fusion Data Foundation subscriptions are required for every OpenShift Container Platform subscribed core with a ratio of 1:1.

Cores versus vCPUs and hyperthreading

Making a determination about whether or not a particular system consumes one or more cores is currently dependent on whether or not that system has hyperthreading available. Hyperthreading is only a feature of Intel CPUs.

For systems where hyperthreading is enabled and where one hyperthread equates to one visible system core, the calculation of cores is a ratio of 2 cores to 4 vCPUs. Therefore, a 2-core subscription covers 4 vCPUs in a hyperthreaded system. A large virtual machine (VM) might have 8 vCPUs, equating to 4 subscription cores. As subscriptions come in 2-core units, you need two 2-core subscriptions to cover these 4 cores or 8 vCPUs.

Where hyperthreading is not enabled, and where each visible system core correlates directly to an underlying physical core, the calculation of cores is a ratio of 2 cores to 2 vCPUs.

Visit IBM Support to determine whether a particular system supports hyperthreading.

- **Cores versus vCPUs and simultaneous multithreading (SMT) for IBM Power**
  Making a determination about whether or not a particular system consumes one or more cores is currently dependent on the level of simultaneous multithreading configured (SMT).

Cores versus vCPUs and simultaneous multithreading (SMT) for IBM Power

Making a determination about whether or not a particular system consumes one or more cores is currently dependent on the level of simultaneous multithreading configured (SMT).

IBM Power provides simultaneous multithreading levels of 1, 2, 4 or 8 for each core which correspond to the number of vCPUs as in Table 1.

Table 1. Different SMT levels and their corresponding vCPUs

SMT level	SMT=1	SMT=2	SMT=4	SMT=8
1 Core	# vCPUs=1	# vCPUs=2	# vCPUs=4	# vCPUs=8
2 Cores	# vCPUs=2	# vCPUs=4	# vCPUs=8	# vCPUs=16
4 Cores	# vCPUs=4	# vCPUs=8	# vCPUs=16	# vCPUs=32

For systems where SMT is configured the calculation for the number of cores required for subscription purposes depends on the SMT level. Therefore, a 2-core subscription corresponds to 2 vCPUs on SMT level of 1, and to 4 vCPUs on SMT level of 2, and to 8 vCPUs on SMT level of 4 and to 16 vCPUs on SMT level of 8 as seen in the preceding table. A large virtual machine (VM) might have 16 vCPUs, which at a SMT level 8 requires a 2 core subscription based on dividing the # of vCPUs by the SMT level (16 vCPUs / 8 for SMT-8 = 2). As subscriptions come in 2-core units, you need one 2-core subscription to cover these 2 cores or 16 vCPUs.

Splitting cores

Systems that require an odd number of cores need to consume a full 2-core subscription. For example, a system that is calculated to require only 1 core ends up consuming a full 2-core subscription once it is registered and subscribed.

When a single virtual machine (VM) with 2 vCPUs uses hyperthreading resulting in 1 calculated vCPU, a full 2-core subscription is required; a single 2-core subscription may not be split across two VMs with 2 vCPUs using hyperthreading. For more information, see Cores versus vCPUs and hyperthreading.

It is recommended that virtual instances be sized so that they require an even number of cores.

- **Shared processor pools for IBM Power**
  IBM Power has a notion of shared processor pools. The processors in a shared processor pool can be shared across the nodes in the cluster. The aggregate compute capacity that is required for IBM Fusion Data Foundation should be a multiple of core-pairs.

Shared processor pools for IBM Power

IBM Power has a notion of shared processor pools. The processors in a shared processor pool can be shared across the nodes in the cluster. The aggregate compute capacity that is required for IBM Fusion Data Foundation should be a multiple of core-pairs.

Subscription requirements

IBM Fusion Data Foundation components can run on either OpenShift Container Platform worker or infrastructure nodes, for which you can use either Red Hat CoreOS (RHCOS) or Red Hat Enterprise Linux (RHEL) 8.4 as the host operating system. RHEL 7 is now deprecated. Fusion Data Foundation subscriptions are required for every OpenShift Container Platform subscribed core with a ratio of 1:1.

When using infrastructure nodes, the rule to subscribe all OpenShift worker node cores for Fusion Data Foundation applies even though they do not need any OpenShift Container Platform or any Fusion Data Foundation subscriptions. You can use labels to state whether a node is a worker or an infrastructure node.

For more information, see How to use dedicated worker nodes for IBM Fusion Data Foundation.

Infrastructure requirements

Understand the infrastructure requirements when planning for Fusion Data Foundation.

- **Platform requirements**
  IBM Fusion Data Foundation 4.21 is supported only on OpenShift Container Platform version 4.21 and its next minor versions.
- **Resource requirements**
  IBM Fusion Data Foundation services consist of an initial set of base services, and can be extended with additional device sets. All of these Fusion Data Foundation services pods are scheduled by Kubernetes on OpenShift Container Platform nodes. Expanding the cluster in multiples of three, one node in each failure domain, is an easy way to satisfy the pod placement rules.
- **Pod placement rules**
  Kubernetes is responsible for pod placement based on declarative placement rules.
- **Storage device requirements**
  Understand the different storage capacity requirements that should be considered when planning internal mode deployments and upgrades.

Platform requirements

IBM Fusion Data Foundation 4.21 is supported only on OpenShift Container Platform version 4.21 and its next minor versions.

Bug fixes for previous version of Fusion Data Foundation are released as bug fix versions.

For external cluster subscription requirements, see the Which Red Hat subscriptions do I need to get Red Hat OpenShift Container Storage support? Red Hat Knowledgebase solution on the Red Hat Customer Portal.

- **Amazon EC2**
  Amazon EC2 supports internal IBM Fusion Data Foundation clusters only.
- **Bare Metal**
  Bare metal supports internal clusters and consuming external clusters.
- **VMware vSphere**
  VMware vSphere supports internal clusters and consuming external clusters.
- **Microsoft Azure**
  Microsoft Azure supports internal IBM Fusion Data Foundation clusters only.
- **Google Cloud**
  Google Cloud supports internal IBM Fusion Data Foundation clusters only.
- **Red Hat OpenStack Platform [Technology Preview]**
  Red Hat OpenStack Platform supports internal IBM Fusion Data Foundation clusters and consuming external clusters.
- **IBM Power**
  IBM Power supports internal IBM Fusion Data Foundation clusters and consuming external clusters.
- **IBM Z and IBM LinuxONE**
  IBM Z and IBM LinuxONE support internal IBM Fusion Data Foundation clusters. Also, supports external mode where IBM Storage Ceph is running on x86.

Amazon EC2

Amazon EC2 supports internal IBM Fusion Data Foundation clusters only.

An Internal cluster must meet both, storage device requirements and have a storage class that provides, EBS storage via the `aws-ebs` provisioner.

Fusion Data Foundation supports `gp2-csi` and `gp3-csi` drivers that were introduced by Amazon Web Services (AWS). These drivers offer better storage expansion capabilities and a reduced monthly price point (`gp3-csi`). You can now select the new drivers when selecting your storage class. In case a high throughput is required, `gp3-csi` is recommended to be used when deploying Fusion Data Foundation.

If you need a high input/output operation per second (IOPS), the recommended EC2 instance types are `D2` or `D3`.

Bare Metal

Bare metal supports internal clusters and consuming external clusters.

An internal cluster must meet both the storage device requirements and have a storage class that provide local SSD (NVMe/SATA/SAS, SAN) via the Local Storage Operator.

VMware vSphere

VMware vSphere supports internal clusters and consuming external clusters.

Recommended versions:

- vSphere 7.0 or later
- vSphere 8.0 or later

For more information, see Installing > Installing on vSphere > Preparing to install on vSphere > VMware vSphere infrastructure requirements within the Red Hat OpenShift Container Platform product documentation.
Note: If VMware ESXi does not recognize its devices as flash, mark them as flash devices. Before IBM Fusion Data Foundation deployment, refer to Mark Storage Devices as Flash.
Additionally, an internal cluster must meet both storage device requirements and have a storage class providing either,

- vSAN or VMFS datastore via the vsphere-volume provisioner.
- VMDK, RDM, or DirectPath storage devices via the Local Storage Operator.

For more information, see storage device requirements.

Microsoft Azure

Microsoft Azure supports internal IBM Fusion Data Foundation clusters only.

An internal cluster must meet both, storage device requirements and have a storage class that provides, an Azure disk via the azure-disk provisioner.

IBM Fusion Data Foundation supports StorageClass with performance plus enabled. By enabling performance plus, the input/output operations per second (IOPS) and throughput limits can be increased for the following types of disks that are 513 GiB and larger:

- Azure Premium solid-state drives (SSD)
- Standard SSDs

Important: To enable performance plus, you must create a StorageClass with `enablePerformancePlus=True`, and select this StorageClass when deploying Fusion Data Foundation.

Google Cloud

Google Cloud supports internal IBM Fusion Data Foundation clusters only.

An internal cluster must meet both, storage device requirements and have a storage class that provides, a GCE Persistent Disk via the `gce-pd` provisioner.

Red Hat OpenStack Platform [Technology Preview]

Red Hat OpenStack Platform supports internal IBM Fusion Data Foundation clusters and consuming external clusters.

Important: Technology Preview features are not supported with IBM production service level agreements (SLAs), might not be functionally complete, and IBM does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.
An internal cluster must meet both, storage device requirements and have a storage class that provides a standard disk via the Cinder provisioner.

IBM Power

IBM Power supports internal IBM Fusion Data Foundation clusters and consuming external clusters.

An Internal cluster must meet both, storage device requirements and have a storage class providing local SSD (NVMe/SATA/SAS, SAN) via the Local Storage Operator.

IBM Z and IBM LinuxONE

IBM Z and IBM LinuxONE support internal IBM Fusion Data Foundation clusters. Also, supports external mode where IBM Storage Ceph is running on x86.

An Internal cluster must meet both, storage device requirements and have a storage class providing local SSD (NVMe/SATA/SAS, SAN) via the Local Storage Operator.

Resource requirements

IBM Fusion Data Foundation services consist of an initial set of base services, and can be extended with additional device sets. All of these Fusion Data Foundation services pods are scheduled by Kubernetes on OpenShift Container Platform nodes. Expanding the cluster in multiples of three, one node in each failure domain, is an easy way to satisfy the pod placement rules.

For more information, see pod placement rules.
Important: These requirements relate to Fusion Data Foundation services only, and not to any other services, operators, or workloads that are running on these nodes.

Table 1. Aggregate available resource requirements for IBM Fusion Data Foundation only

Deployment Mode	Base services	Additional device Set
Internal	• 30 CPU (logical) • 72 GiB memory • 3 storage devices	• 6 CPU (logical) • 15 GiB memory • 3 storage devices
External	• 4 CPU (logical) • 16 GiB memory	Not applicable

For example, for a three node cluster in an internal mode deployment with a single device set, a minimum of 3 x 10 = 30 units of CPU are required.

For more information, see Subscription Offerings and CPU units.

For additional guidance with designing your IBM Fusion Data Foundation cluster, see the ODF Sizing Tool.
Note:
1 CPU Unit maps to the Kubernetes concept of 1 CPU unit.

- 1 unit of CPU is equivalent to 1 core for non-hyperthreaded CPUs.
- 2 units of CPU are equivalent to 1 core for hyperthreaded CPUs.
- Fusion Data Foundation core-based subscriptions always come in pairs (2 cores).

Table 2. Aggregate minimum resource requirements for IBM Power

Deployment Mode	Base services
Internal	• 48 CPU (logical) • 192 GiB memory • 3 storage devices, each with additional 500GB of disk
External	• 24 CPU (logical) • 48 GiB memory

For example, for a three node cluster in an internal-attached devices mode deployment, a minimum of 3 x 16 = 48 units of CPU and 3 x 64 = 192 GB of memory is required.

- **Resource requirements for IBM Z and IBM LinuxONE infrastructure**
  Use this information to understand resource requirements for IBM Z and IBM LinuxONE infrastructures.
- **Minimum deployment resource requirements**
  An Fusion Data Foundation cluster is deployed with minimum configuration when the standard deployment resource requirement is not met.
- **Compact deployment resource requirements**
  IBM Fusion Data Foundation can be installed on a three-node OpenShift compact bare metal cluster, where all the workloads run on three strong master nodes. There are no worker or storage nodes.
- **Resource requirements for MCG only deployment**
  A Fusion Data Foundation cluster deployed only with the Multicloud Object Gateway (MCG) component provides the flexibility in deployment and helps to reduce the resource consumption.
- **Resource requirements for using Network File system**
  Create exports using Network File System (NFS) that can then be accessed externally from the OpenShift cluster.
- **Resource requirements for performance profiles**
  This section provides information about the resource requirements for the different performance profiles that can be selected at deployment.

Resource requirements for IBM Z and IBM LinuxONE infrastructure

Use this information to understand resource requirements for IBM Z and IBM LinuxONE infrastructures.

Table 1. Aggregate available resource requirements for IBM Fusion Data Foundation only (IBM Z and IBM LinuxONE)

Deployment Mode	Base services	Additional device Set	IBM Z and LinuxONE minimum hardware requirements

Deployment Mode	Base services	Additional device Set	IBM Z and LinuxONE minimum hardware requirements
Internal	• 21 CPU (logical) ◦ 3 nodes with 7 CPUs (logical) each • 72 GiB memory • 3 storage devices	• 3 CPU (logical) • 15 GiB memory • 3 storage devices	1-4 IFL
External	• 4 CPU (logical) • 16 GiB memory	Not applicable	Not applicable

CPU
: Is the number of virtual cores defined in the hypervisor, IBM z/VM, Kernel Virtual Machine (KVM), or both.

IFL (Integrated Facility for Linux)
: Is the physical core for IBM Z and IBM LinuxONE.

Minimum system environment

To operate a minimal cluster, at least one additional IFL is required on top of the entire deployment. Depending on the workload, up to 4 IFLs might be necessary.

Minimum deployment resource requirements

An Fusion Data Foundation cluster is deployed with minimum configuration when the standard deployment resource requirement is not met.

Important considerations for minimum deployment

- For production clusters, deploying Fusion Data Foundation with more than three devices is strongly recommended to improve data resiliency.
- DR installation should not use the minimal deployment configuration, as it can result in installation failure.
- The minimum deployment requirements relate to Fusion Data Foundation services only, and not to any other services, operators or workloads that are running on these nodes.

Table 1. Aggregate resource requirements for Fusion Data Foundation only

Deployment Mode	Base services
Internal	• 24 CPU (logical) 15 CPUs for IBM Z and IBM LinuxONE • 72 GiB memory • 3 storage devices

If you want to add additional device sets, convert your minimum deployment to standard deployment.

Compact deployment resource requirements

IBM Fusion Data Foundation can be installed on a three-node OpenShift compact bare metal cluster, where all the workloads run on three strong master nodes. There are no worker or storage nodes.

Important: These requirements relate to Fusion Data Foundation services only, and not to any other services, operators or workloads that are running on these nodes.

Table 1. Aggregate resource requirements for Fusion Data Foundation only

Deployment Mode	Base services	Additional device Set
Internal	• 24 CPU (logical) • 72 GiB memory • 3 storage devices	• 6 CPU (logical) • 15 GiB memory • 3 storage devices

To configure OpenShift Container Platform on a compact bare metal cluster, see Configuring a three-node cluster and Delivering a Three-node Architecture for Edge Deployments.

Resource requirements for MCG only deployment

A Fusion Data Foundation cluster deployed only with the Multicloud Object Gateway (MCG) component provides the flexibility in deployment and helps to reduce the resource consumption.

Table 1. Aggregate resource requirements for MCG only deployment

Deployment Mode	Core	Database (DB)	Endpoint
Internal	• 1 CPU • 4 GiB memory	• 1 CPU • 4 GiB memory	• 1 CPU • 2 GiB memory Note: The default auto scale is between 1 - 2.

Resource requirements for using Network File system

Create exports using Network File System (NFS) that can then be accessed externally from the OpenShift cluster.

You can create exports using Network File System (NFS) that can then be accessed externally from the OpenShift cluster. If you plan to use this feature, the NFS service consumes 3 CPUs and 8Gi of RAM. NFS is optional and is disabled by default.

The NFS volume can be accessed in two ways:

- In-cluster: by an application pod inside of the OpenShift cluster.
- Out of cluster: from outside of the OpenShift cluster.

For more information about the NFS feature, see Creating exports using NFS.

Resource requirements for performance profiles

This section provides information about the resource requirements for the different performance profiles that can be selected at deployment.

Fusion Data Foundation provides three performance profiles to enhance the performance of the clusters. You can choose one of these profiles based on your available resources and desired performance level during deployment or post deployment.

Table 1. Recommended resource requirements for different performance profiles

Performance profile	CPU	CPU on IBM Z or LinuxONE	Memory
Lean	24	15	72 GiB
Balanced	30	21	72 GiB
Performance	45	30	96 GiB

Important: Ensure to select the profiles based on the available free resources as you might already be running other workloads.

Pod placement rules

Kubernetes is responsible for pod placement based on declarative placement rules.

The IBM Fusion Data Foundation base service placement rules for Internal cluster can be summarized as follows:

- Nodes are labeled with the `cluster.ocs.openshift.io/openshift-storage` key.
- Nodes are sorted into pseudo failure domains if none exist.
- Components requiring high availability are spread across failure domains.
- A storage device must be accessible in each failure domain.

This leads to the requirement that there be at least three nodes, and that nodes be in three distinct rack or zone failure domains in the case of pre-existing topology labels.

For additional device sets, there must be a storage device, and sufficient resources for the pod consuming it in each of the three failure domains. Manual placement rules can be used to override default placement rules, but generally this approach is only suitable for bare metal deployments.

Storage device requirements

Understand the different storage capacity requirements that should be considered when planning internal mode deployments and upgrades.

The general recommendation is 12 devices or less per node. This recommendation ensures that nodes stay below cloud provider dynamic storage device attachment limits, and to limit the recovery time after node failures with local storage devices. Expanding the cluster in multiples of three, one node in each failure domain, is an easy way to satisfy pod placement rules.

Storage nodes should have at least two disks, one for the operating system and the remaining disks for Fusion Data Foundation components.
Note: You can expand the storage capacity only in the increment of the capacity that is selected at the time of installation.

- **Dynamic storage devices**
  IBM Fusion Data Foundation permits the selection of either 0.5 TiB, 2 TiB or 4 TiB capacities as the request size for dynamic storage device sizes.
- **Local storage devices**
  For local storage deployment, any disk size of 16 TiB or less can be used, and all disks should be of the same size and type. The number of local storage devices that can run per node is a function of the node size and resource requirements.
- **Capacity planning**
  Always ensure that available storage capacity stays ahead of consumption. Recovery is difficult if available storage capacity is completely exhausted, and requires

more intervention than adding capacity, deleting, or migrating content alone.

Dynamic storage devices

IBM Fusion Data Foundation permits the selection of either 0.5 TiB, 2 TiB or 4 TiB capacities as the request size for dynamic storage device sizes.

The number of dynamic storage devices that can run per node is a function of the node size, underlying provisioner limits and resource requirements.

Local storage devices

For local storage deployment, any disk size of 16 TiB or less can be used, and all disks should be of the same size and type. The number of local storage devices that can run per node is a function of the node size and resource requirements.

For more information, see resource requirements. Expanding the cluster in multiples of three, one node in each failure domain, is an easy way to satisfy pod placement rules.
Note: Disk partitioning is not supported.

Capacity planning

Always ensure that available storage capacity stays ahead of consumption. Recovery is difficult if available storage capacity is completely exhausted, and requires more intervention than adding capacity, deleting, or migrating content alone.

Capacity alerts are issued when cluster storage capacity reaches 75% (near-full) and 85% (full) of total capacity. Always address capacity warnings promptly, and review your storage regularly to ensure that you do not run out of storage space. When you get to 75% (near-full), either free up space or expand the cluster. When you get the 85% (full) alert, it indicates that you have run out of storage space completely and cannot free up space using standard commands. If this occurs, contact IBM Support.

Table 1 and Table 2 show example node configurations for Fusion Data Foundation with dynamic storage devices.

Table 1. Example initial configurations with 3 nodes

Storage Device size	Storage Devices per node	Total capacity	Usable storage capacity
0.5 TiB	1	1.5 TiB	0.5 TiB
2 TiB	1	6 TiB	2 TiB
4 TiB	1	12 TiB	4 TiB

Table 2. Example of expanded configurations with 30 nodes (N)

Storage Device size (D)	Storage Devices per node (M)	Total capacity (D * M * N)	Usable storage capacity (D*M*N/3)
0.5 TiB	3	45 TiB	15 TiB
2 TiB	6	360 TiB	120 TiB
4 TiB	9	1080 TiB	360 TiB

Network requirements

Use this section to understand the different network considerations when planning deployments.

- **IPv6 support**
  IBM Fusion Data Foundation supports IPv6 in single stack.
- **Multi network plug-in (Multus) support**
  Multi-network plug-in (Multus) support allows one or more host network interfaces to be reserved for exclusive use by Fusion Data Foundation on bare metal infrastructures to improve security and performance through traffic isolation.

IPv6 support

IBM Fusion Data Foundation supports IPv6 in single stack.

Fusion Data Foundation supports IPv6. This IPv6 is supported in single stack only, and cannot be used simultaneously with IPv4. IPv6 is the default behavior in Fusion Data Foundation when IPv6 is turned on in OpenShift Container Platform.

IBM Fusion Data Foundation also supports IPv6 auto detection and configuration. Clusters using IPv6 are automatically configured accordingly.

OpenShift Container Platform dual stack with IBM Fusion Data Foundation IPv4 is supported, however Dual stack on IBM Fusion Data Foundation IPv6 is not supported.

Multi network plug-in (Multus) support

Multi-network plug-in (Multus) support allows one or more host network interfaces to be reserved for exclusive use by Fusion Data Foundation on bare metal infrastructures to improve security and performance through traffic isolation.

IBM Fusion Data Foundation supports the ability to use multi-network plug-in (Multus) on bare metal infrastructures to improve security and performance by isolating the different types of network traffic. By using Multus, one or more network interfaces on hosts can be reserved for exclusive use of Fusion Data Foundation. For more details, Multus architecture for Fusion Data Foundation.

Disaster Recovery

Disaster Recovery (DR) helps an organization to recover and resume business critical functions or normal operations when there are disruptions or disasters.

Fusion Data Foundation provides High Availability (HA) & DR solutions for stateful apps which are broadly categorized into two broad categories:

- **Metro-DR**: Single Region and cross data center protection with no data loss.
- **Disaster Recovery with stretch cluster**: Single OpenShift Data Foundation cluster is stretched between two different locations to provide the storage infrastructure with disaster recovery capabilities.
- **Regional-DR** : Cross Region protection with minimal potential data loss.

- **Metro-DR**
  Metro-DR is composed of Red Hat Advanced Cluster Management for Kubernetes (RHACM), IBM Storage Ceph and Fusion Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters.
- **Disaster Recovery with stretch cluster**
  Stretch cluster disaster recovery spans a single OpenShift cluster across two data zones with a third arbiter zone, providing resilience against zone failure with a maximum round-trip latency of 10 ms between data zones.
- **Regional-DR**
  Regional disaster recovery (Regional-DR) is composed of Red Hat Advanced Cluster Management for Kubernetes (RHACM) and IBM Fusion Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters. It is built on Asynchronous data replication and hence could have a potential data loss but provides the protection against a broad set of failures.

Metro-DR

Metro-DR is composed of Red Hat Advanced Cluster Management for Kubernetes (RHACM), IBM Storage Ceph and Fusion Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters.

This release of Metro-DR solution provides volume persistent data and metadata replication across sites that are geographically dispersed. In the public cloud these would be similar to protecting from an Availability Zone failure. Metro-DR ensures business continuity during the unavailability of a data center with no data loss.
Important: You can now set up Metropolitan disaster recovery solutions for workloads based on OpenShift virtualization technology using Fusion Data Foundation. For more information, see the Use OpenShift Data Foundation Disaster Recovery to Protect Virtual Machines Red Hat knowledgebase article.
Prerequisites

Ensure that the primary managed cluster (Site-1) is co-situated with the active RHACM hub cluster while the passive hub cluster is situated along with the secondary managed cluster (Site-2). Alternatively, the active RHACM hub cluster can be placed in a neutral site (Site-3) that is not impacted by the failures of either of the primary managed cluster at Site-1 or the secondary cluster at Site-2. In this situation, if a passive hub cluster is used it can be placed with the secondary cluster at Site-2.
Note: Hub recovery is a Technology Preview feature and is subject to Technology Preview support limitations.
For detailed solution requirements, see the following:

- Configuring Data Foundation Metro-DR
- Install > Installing > Requirements and recommendations within Red Hat Advanced Cluster Management for Kubernetes product documentation.

Disaster Recovery with stretch cluster

Stretch cluster disaster recovery spans a single OpenShift cluster across two data zones with a third arbiter zone, providing resilience against zone failure with a maximum round-trip latency of 10 ms between data zones.

In this case, a single cluster is stretched across two zones with a third zone as the location for the arbiter. This feature is currently intended for deployment in the OpenShift Container Platform on-premises and in the same data center. Consider Metro-DR as a first option for no data loss DR solution deployed over multiple data centers with low latency networks.

Note: The stretch cluster solution is designed for deployments where latencies do not exceed 10 ms maximum round-trip time (RTT) between the zones containing data volumes. For Arbiter nodes follow the latency requirements specified for etcd, see the Guidance for Red Hat OpenShift Container Platform Clusters - Deployments Spanning Multiple Sites(Data Centers/Regions) Red Hat knowledgebase article. Contact IBM Support if you are planning to deploy with higher latencies.
To use the stretch cluster

- You must have a minimum of five nodes across three zones, where two nodes per zone are used for each data-center zone, and one additional zone with one node is used for arbiter zone (the arbiter can be on a master node).
- All the nodes must be manually labeled with the zone labels prior to cluster creation.
  For example, the zones can be labeled as:
  - topology.kubernetes.io/zone=arbiter (master or worker node)
  - topology.kubernetes.io/zone=datacenter1 (minimum two worker nodes)
  - topology.kubernetes.io/zone=datacenter2 (minimum two worker nodes)

For more information, see Disaster recovery with stretch cluster for Fusion Data Foundation.

Regional-DR

Regional disaster recovery (Regional-DR) is composed of Red Hat Advanced Cluster Management for Kubernetes (RHACM) and IBM Fusion Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters. It is built on Asynchronous data replication and hence could have a potential data loss but provides the protection against a broad set of failures.

Fusion Data Foundation is backed by Ceph as the storage provider, whose lifecycle is managed by Rook and it is enhanced with the ability to:

- Enable pools for mirroring.
- Automatically mirror images across RBD pools.
- Provides csi-addons to manage per Persistent Volume Claim mirroring.

Regional-DR supports Multi-Cluster configuration that is deployed across different regions and data centers. For example, a 2-way replication across two managed clusters located in two different regions or data centers. This solution is entitled with Red Hat Advanced Cluster Management (RHACM) and Fusion Data Foundation Advanced SKUs and related bundles.

Prerequisites

- Disaster Recovery features supported by Fusion Data Foundation require that Fusion subscription be active on both source and destination clusters in order to successfully implement a Disaster Recovery solution.
- Ensure that the primary managed cluster (Site-1) is co-situated with the active RHACM hub cluster while the passive hub cluster is situated along with the secondary managed cluster (Site-2). Alternatively, the active RHACM hub cluster can be placed in a neutral site (Site-3) that is not impacted by the failures of either of the primary managed cluster at Site-1 or the secondary cluster at Site-2. In this situation, if a passive hub cluster is used it can be placed with the secondary cluster at Site-2.
  Note: Hub recovery is a Technology Preview feature and is subject to Technology Preview support limitations.

For detailed solution requirements, see Requirements for enabling Regional-DR and Install > Installing > Requirements and recommendations within Red Hat Advanced Cluster Management for Kubernetes product documentation.

Supported and unsupported features for IBM Power and IBM Z infrastructures

Use this information to understand which features are supported and unsupported in IBM Power and IBM Z infrastructures.

Table 1 lists the supported and unsupported features on IBM Power and IBM Z infrastructures.

Table 1. Supported and unsupported features on IBM Power and IBM Z infrastructures

Features	IBM Power	IBM Z infrastructure
Compact deployment	Unsupported	Unsupported
Dynamic storage devices	Unsupported	Supported
Stretched Cluster - Arbiter	Supported	Unsupported
Federal Information Processing Standard Publication (FIPS)	Unsupported	Unsupported
Ability to view pool compression metrics	Supported	Unsupported
Automated scaling of Multicloud Object Gateway (MCG) endpoint pods	Supported	Unsupported
Alerts to control overprovision	Supported	Unsupported
Alerts when Ceph Monitor runs out of space	Supported	Unsupported
Deployment of standalone Multicloud Object Gateway component	Supported	Unsupported
Extended Fusion Data Foundation control plane which allows pluggable external storage such as IBM Flashsystem	Unsupported	Unsupported
IPV6 support	Unsupported	Unsupported
Multus	Unsupported	Unsupported
Multicloud Object Gateway (MCG) bucket replication	Supported	Unsupported
Quota support for object data	Supported	Unsupported
Minimum deployment	Unsupported	Unsupported
Regional-Disaster Recovery (Regional-DR) with Red Hat Advanced Cluster Management (RHACM)	Supported	Unsupported
Metro-Disaster Recovery (Metro-DR) multiple clusters with RHACM	Supported	Supported
Support for network file system (NFS) services	Supported	Unsupported
Ability to change Multicloud Object Gateway (MCG) account credentials	Supported	Unsupported
Multicluster monitoring in Red Hat Advanced Cluster Management console	Supported	Unsupported
Deletion of expired objects in Multicloud Object Gateway lifecycle	Supported	Unsupported
Agnostic deployment of OpenShift Data Foundation on any Openshift supported platform	Unsupported	Unsupported
Installer provisioned deployment of OpenShift Data Foundation using bare metal infrastructure	Unsupported	Unsupported
Openshift dual stack with OpenShift Data Foundation using IPv4	Unsupported	Unsupported
Ability to disable Multicloud Object Gateway external service during deployment	Unsupported	Unsupported
Ability to allow overriding of default NooBaa backing store	Supported	Unsupported
Allowing ocs-operator to deploy two MGR pods, one active and one standby	Unsupported	Unsupported
Disaster Recovery for brownfield deployments	Unsupported	Supported

Installing Fusion Data Foundation

Install Fusion Data Foundation in local, dynamic, external, client, or provider modes.

Before you begin

- Important: If you install Fusion Data Foundation 4.21 or later and the Global Data Platform remote mount service is installed and configured, the service is automatically migrated to version 6.0.1.0. Review the supported migration scenarios before installation.
- Existing storage installations
  If you already have an installation of Fusion Data Foundation cluster or Red Hat® OpenShift® Data Foundation cluster, the IBM Fusion UI automatically discovers it.
  - In the Data Foundation page, you can only view the Usable capacity, Health, and information about the storage nodes.
  - If it is Red Hat OpenShift Data Foundation, migrate it to Fusion Data Foundation by following the instructions in Upgrading Red Hat OpenShift Data Foundation to IBM Storage Fusion Data Foundation.
  - If you have a Rook-Ceph operator, contact IBM Support for guidance on expected behavior.
- Storage device and node requirements
  - Use a maximum of nine storage devices per node to ensure optimal recovery time and stay within cloud provider dynamic storage device attachment limits. Using more storage devices increases recovery time if a node is lost.
  - Add nodes in multiples of three, with each node in a different failure domain.
  - For deployments with three failure domains, scale up storage by adding disks in multiples of three, with the same number of disks from nodes in each failure domain.
- Clean installation verification
  Before enabling the Fusion Data Foundation service, verify that there is no previously installed Fusion Data Foundation, Red Hat OpenShift Data Foundation, OpenShift Container Storage, or Rook-Ceph in this OpenShift cluster:
  1. Verify that the namespace `openshift-storage` does not exist.
  2. Run the following command to confirm that no storage resources exist:
     ```
     oc get storagecluster -A; oc get cephcluster -A
     ```
  3. Run the following command to confirm that no nodes have storage labels:
     ```
     oc get node -l cluster.ocs.openshift.io/openshift-storage=
     ```
- Infrastructure nodes
  Ensure that the following prerequisites are met:
  - For infrastructure requirements, see Infrastructure requirements.
  - For Fusion Data Foundation resource requirements, see ODF sizer tool. For common Fusion Data Foundation components per instance requirements of CPU and Memory, see Resource requirements.
  - If you want to deploy Fusion Data Foundation in an Infra node, make sure the worker node label exists.
    ```
    oc get node -l "node-role.kubernetes.io/worker="
    ```
  - Ensure that there are no taints in the infra or compute nodes that are used as storage nodes.

  For more information about the limitations and considerations of using OVE as an alternative to OpenShift Container Platform platform, see OVE considerations.

About this task

- Both Data Foundation service and Global Data Platform service can coexist during the following scenarios:
  - Deploy Data Foundation service without `dedicated` mode.
  - Deploy Data Foundation in `dedicated` mode first, and then deploy Global Data Platform service.
- For Red Hat OpenShift Data Foundation deployed outside of IBM Fusion, you cannot add nodes or capacity from IBM Fusion. See Scaling storage in *Red Hat OpenShift Data Foundation* documentation.
- For client cluster with Fusion Data Foundation, you do not install IBM Fusion spoke cluster or Fusion Data Foundation from the user interface. These installations are automatically run from the IBM Fusion HCI hub cluster.
- Data Foundation service supports five storage types: local, dynamic, external, client, and provider. Only one device type can be used for Data Foundation service in an OpenShift Container Platform. For more information about the different deployment modes, see Storage cluster deployment approaches.

  The following table shows supported platforms and which device types are supported in each platform.

Platform	Support
VMware	Local, dynamic, client, and external device
Bare Metal	Local, external device, client, provider mode
Linux on IBM Z	Local, dynamic, client, and external device
IBM Power Systems	Local, client, and external device
ROKS on IBM Cloud (ROKS on VPC and Classic Bare Metal)	Cloud-managed Fusion Data Foundation service. For the procedure to install, see Understanding Fusion Data Foundation.
Self managed OpenShift Container Platform on Microsoft Azure	Dynamic and client
Self managed OpenShift Container Platform on Amazon Web Services	Dynamic and client
Amazon Web Services ROSA and Amazon Web Services ROSA HCP	Dynamic and client
Self managed OpenShift Container Platform on Google Cloud	Dynamic and client

- For IBM Fusion services and platform support matrix, see IBM Fusion services and platform support matrix.

Procedure

1. Go to Services page in IBM Fusion web console.
2. In the Available section, click the Data Foundation tile.
3. In the Data Foundation page, go through the details about the service and click Install.
4. In the Install service window, select the Device type. The available options can be Local, Dynamic, or External based on which option is supported on your platform. If you want to use storage through a provider and work with Data Foundation service storage as a client, then there is no Install button available. A notification is displayed asking you to connect to the provider cluster.

5. Choose whether to enable Automatic updates and click Install.
   Note: If you select Automatic updates, then IBM Fusion auto upgrades Fusion Data Foundation version.
   For example, 4.14.x to 4.14.y, 4.16.x to 4.16.y, and across the subscription channels like 4.14.x to 4.15.y, 4.15.x to 4.16.x. For storage support matrix, see Support matrix.

   However, it does not auto convert Red Hat OpenShift Data Foundation to Fusion Data Foundation.

   The Fusion Data Foundation operator starts to deploy and you can see Data Foundation in the Installed section of the Services page. Initially, the status shows as Installing and progress of the installation is mentioned in percentage. After successful completion of the installation, the status changes to Running. A Data Foundation page is added to the menu.

6. To validate the installation, do the following steps in OpenShift Container Platform web console:
   - Go to Installed Operators to check whether the Fusion Data Foundation operator is listed and Status shows as Succeeded.
   - Additionally, if the installed platform is Bare Metal, Linux on IBM Z, or VMware using local device, check whether the `Local Storage operators` is installed and its status shows as Succeeded.
7. Verify the Fusion Data Foundation service installation.
   a. To verify the installation of Fusion Data Foundation on IBM Fusion, perform the checks specified in Verify the Fusion Data Foundation.
   b. Additionally, run the command to verify that Fusion Data Foundation service is installed successfully:

      ```
      oc describe odfmanager/odfmanager -n <Fusionns>
      ```

      Update **<Fusionns>** with your namespace value.

Table 1. Install states Fusion Data Foundation service

Install State	Description
`Installing`	Fusion Data Foundation installation is ongoing, and there are no errors yet.
`Failing`	There is an installation error but IBM Fusion retries to install Fusion Data Foundation. If the problem is solved, this status changes to `Installing` or `Completed`.
`Completed`	Fusion Data Foundation installation completed successfully.

Table 2. Upgrade states Fusion Data Foundation service

Upgrade State	Description
`Not started`	Fusion Data Foundation upgrade has not started yet.
`Upgrading`	Fusion Data Foundation upgrade is ongoing, and there are no errors yet.
`Failing`	There is an installation error but IBM Fusion retries to upgrade Fusion Data Foundation. If the problem is solved, this status changes to `Upgrading` or `Completed`.
`Completed`	Fusion Data Foundation upgrade completed successfully.

Table 3. Health states Fusion Data Foundation service

State	Description
`Installing`	Fusion Data Foundation installation is ongoing. For more information about installation status details, see Table 1.
`Upgrading`	Fusion Data Foundation upgrade is ongoing. For more information about upgrading status details, see Table 2.
`Healthy`	Fusion Data Foundation installation completed successfully and the service is in normal state.
`Degraded`	Fusion Data Foundation installation completed successfully but the service is not in normal state.

What to do next

Configure the Fusion Data Foundation storage service by following the instructions mentioned in Configuring Fusion Data Foundation.

Optionally, you can also setup encryption to use an external Key Management System. For the procedure to set up encryption, see Preparing to connect to an external KMS server in Fusion Data Foundation.

Note: You can view the Usable capacity and Health for an externally discovered Fusion Data Foundation but cannot configure storage.

Uninstalling Fusion Data Foundation

Use the following steps to uninstall Data Foundation by using command line interface.

Before you begin

- Ensure that all workloads are stopped. Delete the associated persistent volume claims (PVCs) and clean up the namespace linked to it. Also, remove any other IBM Fusion services that use the Fusion Data Foundation service.
- Reclaim access to the data stored in the Fusion Data Foundation cluster.
- If you are uninstalling Fusion Data Foundation 4.21 or later on Red Hat® OpenShift® Container Platform 4.21 or later, uninstall the Global Data Platform remote mount service if it is installed. For instructions, see Uninstalling Global Data Platform from Fusion Data Foundation external mode.

Procedure

1. Run the following command and export IBM Fusion namespace as environmental variable.

```
export FUSION_NS="<Fusion namespace>"
```

Important: Replace `<Fusion namespace>` with your namespace.

2. Run the following **oc** command and scale the IBM Fusion storage deployment to 0 replica.

```
oc scale deployment --replicas=0 isf-cns-operator-controller-manager -n "$FUSION_NS"
```

3. Run the following commands and delete the Fusion Data Foundation CR.

```
oc delete odfmanager odfmanager
oc delete odfcluster odfcluster -n "$FUSION_NS"
```

4. Uninstall the Fusion Data Foundation. For uninstallation steps, see Uninstalling Data foundation in Internal mode deployment on the IBM Support page.
   Important: Ensure that the `openshift-storage` namespace and all associated resources are deleted before you proceed to the next step. For external mode deployments, follow the internal mode procedure Uninstalling Data foundation in Internal mode deployment described in the IBM Support page to remove the namespace and associated resources.
5. Run the following **oc** command and scale the IBM Fusion storage deployment to 1 replica.

```
oc scale deployment --replicas=1 isf-cns-operator-controller-manager -n "$FUSION_NS"
```

6. Run the following **oc** command to delete the Fusion Data Foundation Service Instance CR.

```
oc delete fusionserviceinstance odfmanager -n "$FUSION_NS"
```

7. Run the following **oc** command to delete the Fusion Data Foundation Catalog Source.

```
oc delete catalogsource isf-data-foundation-catalog -n openshift-marketplace
```

- **Uninstalling Global Data Platform from Fusion Data Foundation external mode**
  Uninstall Global Data Platform from Fusion Data Foundation external mode by removing associated IBM Storage Scale configurations and resources.

Uninstalling Global Data Platform from Fusion Data Foundation external mode

Uninstall Global Data Platform from Fusion Data Foundation external mode by removing associated IBM Storage Scale configurations and resources.

Before you begin

- Ensure that all applications that access the storage filesystem through the IBM Storage Scale Container Storage Interface Driver (CSI) are stopped.
- Do the following steps to remove all applications:
  1. Stop all applications that access storage by using the IBM Storage Scale CSI driver.
  2. Delete all Persistent Volume Claims (PVCs) and Persistent Volumes (PVs) created by the IBM Storage Scale CSI driver.

About this task

This procedure applies to uninstalling Global Data Platform from Fusion Data Foundation in external mode by removing IBM Storage Scale configurations and resources on Red Hat® OpenShift® Container Platform 4.21 or later and Fusion Data Foundation 4.21 or later.

For Global Data Platform uninstallation on Red Hat OpenShift Container Platform 4.20 or earlier and Fusion Data Foundation 4.20 or earlier, see Uninstalling Global Data Platform.

Procedure

1. Set the namespace.
   Command example:

```
export SCALE_NS="ibm-spectrum-scale"
```

2. Check for existing `VolumeSnapshot` objects.
   Command example:

```
oc get volumesnapshotclass | grep spectrumscale.csi.ibm.com
```

```
for VSC in $(oc get volumesnapshotclass -o name | grep spectrumscale.csi.ibm.com | awk -F/ '{print $2}'); do
  oc get volumesnapshot --all-namespaces -o=jsonpath="{range .items[?(@.spec.volumeSnapshotClassName=='$VSC')]}
{.metadata.namespace}:{.metadata.name}{'\n'}{end}"
done
```

   If any volume snapshots are returned, delete them before continuing.

3. Check for existing PVCs and PVs.
   Command example:

```
oc get sc | grep spectrumscale.csi.ibm.com
```

```
for SC in $(oc get sc -o name | grep spectrumscale.csi.ibm.com | awk -F/ '{print $2}'); do
  oc get pvc --all-namespaces -o=jsonpath="{range .items[?(@.spec.storageClassName=='$SC')]}{.metadata.namespace}:
{.metadata.name}{'\n'}{end}"
  oc get pv -o=jsonpath="{range .items[?(@.spec.storageClassName=='$SC')]}{.metadata.name}{'\n'}{end}"
done
```

   If any PVCs or PVs exist, delete them before continuing.

4. Delete `EncryptionConfig` custom resources.
   Command example:

```
oc -n $SCALE_NS get EncryptionConfig
oc -n $SCALE_NS get EncryptionConfig -o name | xargs -r oc -n $SCALE_NS delete
```

5. Delete filesystem custom resources.
   Command example:

```
oc -n $SCALE_NS get fs
oc -n $SCALE_NS get fs -o name | xargs -r oc -n $SCALE_NS delete
```

6. Delete `RemoteCluster` custom resources.
   Command example:

```
oc -n $SCALE_NS get remotecluster
oc -n $SCALE_NS get remotecluster -o name | xargs -r oc -n $SCALE_NS delete
```

7. Record the Storage Scale cluster name.
   Command example:

```
oc -n ${SCALE_NS} get daemon -o yaml | grep clusterName: | awk -F " " '{print $2}'
```

   Note: Save the cluster name for later use.
8. Delete the Storage Scale cluster custom resource.
   Command example:

```
oc get cluster.scale.spectrum.ibm.com
oc get cluster.scale.spectrum.ibm.com -o name | xargs -r oc delete
```

9. Delete the `CSIScaleOperator` custom resource.
   Command example:

```
oc -n $SCALE_NS get CSIScaleOperator
oc -n $SCALE_NS get CSIScaleOperator -o name | xargs -r oc -n $SCALE_NS delete
```

10. Scale down the CSV deployment.
    Command example:

```
oc -n $SCALE_NS get csv | grep ibm-spectrum-scale-operator

oc -n $SCALE_NS get csv -o name | grep ibm-spectrum-scale-operator | xargs -r -I{} oc -n $SCALE_NS patch {} --type=json -
p='[{"op":"replace","path":"/spec/install/spec/deployments/0/spec/replicas","value":0}]'
```

11. Delete PMCollector and CSI artifacts.
    Command example:

```
oc -n $SCALE_NS get pvc -o name | grep pmcollector | xargs -r oc -n $SCALE_NS delete

oc delete pv -l app.kubernetes.io/instance=ibm-spectrum-scale,app.kubernetes.io/name=pmcollector

oc delete sc -l app.kubernetes.io/instance=ibm-spectrum-scale,app.kubernetes.io/name=pmcollector
```

12. Delete template PVs.
    Command example:

```
oc get pv -o name | grep 'ibm-spectrum-scale-fs-.*-template' | xargs -r oc delete
```

13. Delete `VolumeSnapshotClass` objects.
    Command example:

```
oc get volumesnapshotclass -o name -o wide | grep spectrumscale.csi.ibm.com
oc get volumesnapshotclass -o jsonpath='{range .items[?(@.driver=="spectrumscale.csi.ibm.com")]}{.metadata.name}{"\n"}
{end}' | xargs -r oc delete volumesnapshotclass
```

14. Delete storage classes.
    Command example:

```
oc get sc -o name | grep spectrumscale.csi.ibm.com
oc get sc -o jsonpath='{range .items[?(@.provisioner=="spectrumscale.csi.ibm.com")]}{.metadata.name}{"\n"}{end}' | xargs -
r oc delete sc
```

15. Delete remote GUI and user detail secrets.
    Command example:

```
oc -n $SCALE_NS get secret
oc -n $SCALE_NS delete secret ibm-spectrum-scale-remote-gui-secret --ignore-not-found
oc -n $SCALE_NS delete secret <clustername-user-details-secret>
```

16. Verify cluster custom resource removal.
    Command example:

```
oc get cluster.scale.spectrum.ibm.com
```

17. Complete the following steps to delete IBM Storage Scale Container Native directories from nodes on which IBM Storage Scale is deployed:
    a. Identify the nodes on which IBM Storage Scale is deployed.
       Command example:

```
kubectl get nodes -o jsonpath="{range .items[*]}{.metadata.name}{'\n'}"
```

       Tip: Use the `nodeSelector` to limit the set of nodes to clean up, or run cleanup on all nodes.
    b. Create a debug pod to remove the kernel modules and host path volume mounted directories used by IBM Storage Scale Container Native.
       Run the following command on each node:

       Command example:

```
oc debug node/<openshift_node> -T -- chroot /host sh -c "rm -rf /var/mmfs; rm -rf /var/adm/ras; rmmod tracedev mmfs26
mmfslinux;"
```

c. Verify that no artifacts remain by running the following validation command:
Command example:

```
oc debug node/<openshift_node> -T -- chroot /host sh -c "ls /var/mmfs; ls /var/adm/ras; rmmod tracedev mmfs26
mmfslinux;"
```

d. Delete IBM Storage Scale node labels created by IBM Storage Scale Container Native operator.
Command example:

```
# clean up all 'scale.spectrum.ibm.com' labels on the nodes
kubectl get nodes -ojson | jq -r '.items[].metadata.labels' | \
grep scale.spectrum.ibm.com | sort | cut -d: -f1 | \
tr -d "\""| uniq | xargs -I{} kubectl label nodes --all {}\-

# clean up 'scale' label on the nodes
oc label node --all scale.spectrum.ibm.com/role-
oc label node --all scale.spectrum.ibm.com/designation-
oc label node --all scale-
```

18. Do the final validation.
Command example:

```
oc get all -n $SCALE_NS
```

Example output:

```
Warning: apps.openshift.io/v1 DeploymentConfig is deprecated in v4.14+, unavailable in v4.10000+
NAME                                                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
service/ibm-spectrum-scale-controller-manager-service   ClusterIP   172.30.47.230    <none>        443/TCP    7d
service/ibm-spectrum-scale-webhook-service              ClusterIP   172.30.204.125   <none>        443/TCP    7d

NAME                                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/ibm-spectrum-scale-controller-manager   0/0     0            0           7d

NAME                                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/ibm-spectrum-scale-controller-manager-57c648dbcc   0         0         0       7d
```

19. Complete the following steps on the remote IBM Storage Scale cluster to delete access permission:
Note: If your storage cluster is running on AWS ROSA, skip this step and revoke filesystem access directly from the storage cluster. For the procedure to revoke, see Cleanup on AWS ROSA.

a. Run the following command to query the name of the containerized client cluster:
Command example:

```
mmauth show all | grep <scale_cluster_name>
```

Replace **<scale_cluster_name>** with the cluster name recorded in step 7.

b. Run the following command to remove the client cluster authorization:
Command example:

```
mmauth delete <scale_cluster_name>
```

Results

Global Data Platform resources are uninstalled, and IBM Storage Scale configurations are removed from Fusion Data Foundation.

What to do next

If a complete cleanup is required, remove the Global Data Platform namespace and related custom resource definitions (CRDs) after the Fusion Data Foundation cleanup is complete. For the Fusion Data Foundation cleanup procedure, see Uninstalling Fusion Data Foundation.

To uninstall the IBM Storage Scale Container Native operator and remove associated Kubernetes objects and namespaces, run the following commands:

```
oc delete project ibm-spectrum-scale --wait=true --timeout=5m
oc delete crd approvalrequests.scale.spectrum.ibm.com cachevolumeoperations.scale.spectrum.ibm.com
cachevolumes.scale.spectrum.ibm.com callhomes.scale.spectrum.ibm.com clusterinterconnects.scale.spectrum.ibm.com
clusters.scale.spectrum.ibm.com compressionjobs.scale.spectrum.ibm.com consistencygroups.scale.spectrum.ibm.com
daemons.scale.spectrum.ibm.com diskjobs.scale.spectrum.ibm.com dnsconfigs.scale.spectrum.ibm.com dnss.scale.spectrum.ibm.com
encryptionconfigs.scale.spectrum.ibm.com filesystemrecoverydata.scale.spectrum.ibm.com filesystems.scale.spectrum.ibm.com
grafanabridges.scale.spectrum.ibm.com guis.scale.spectrum.ibm.com localdisks.scale.spectrum.ibm.com
pmcollectors.scale.spectrum.ibm.com recoveryconfigstores.scale.spectrum.ibm.com recoverygroups.scale.spectrum.ibm.com
regionaldrexports.scale.spectrum.ibm.com regionaldrs.scale.spectrum.ibm.com remoteclusters.scale.spectrum.ibm.com
restripefsjobs.scale.spectrum.ibm.com stretchclusterinitnodes.scale.spectrum.ibm.com stretchclusters.scale.spectrum.ibm.com
stretchclustertiebreakers.scale.spectrum.ibm.com upgradeapprovals.scale.spectrum.ibm.com volumes.scale.spectrum.ibm.com
```

Configuring Fusion Data Foundation

Configure Fusion Data Foundation to set up storage services that support scalable, resilient, and flexible data management for OpenShift® applications.

After the Fusion Data Foundation installation is complete, configure the storage system based on the device type selected during Fusion Data Foundation installation.

Fusion Data Foundation supports the following configuration modes based on device type:

Local
: In local mode, you can deploy Fusion Data Foundation on a smaller scale by using the local storage devices instead of a dedicated storage cluster. It is suitable for edge deployments, testing environments, or smaller setups where a full-scale storage cluster is not required. For instructions, see Configuring Fusion Data Foundation in local mode.

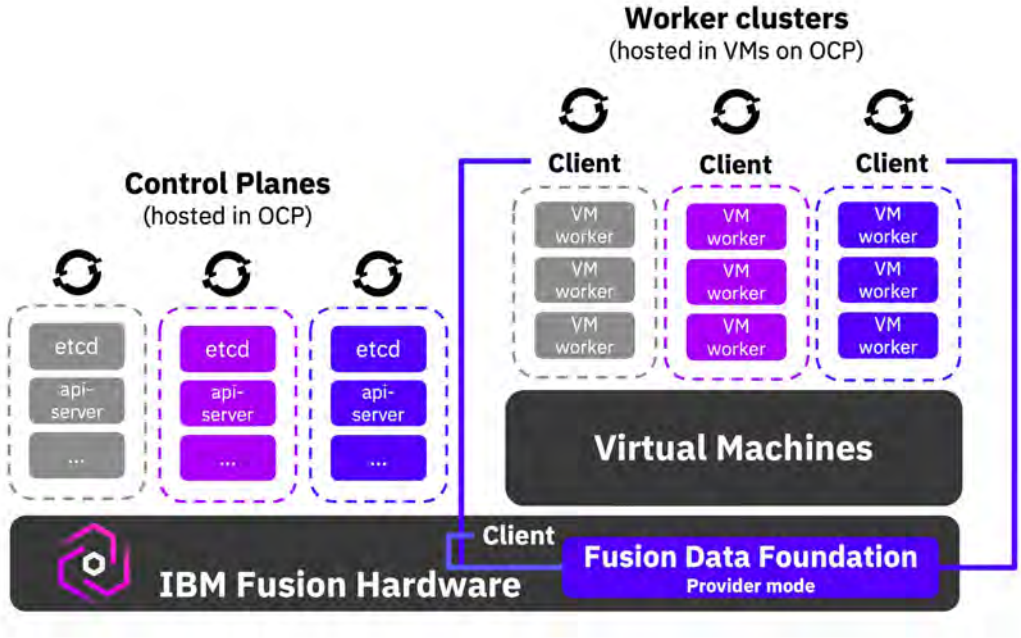
Dynamic
: The dynamic mode is designed for environments where existing storage is available. In this mode, you can use an existing storage provider on a OpenShift cluster and build a Fusion Data Foundation cluster or a fault tolerant storage cluster on top of it. This mode integrates with external storage solutions to provide scalable, resilient, and dynamically managed storage resources for your OpenShift cluster. For instructions, see Configuring Fusion Data Foundation in dynamic mode.

External
: The external mode is designed for scenarios where the storage infrastructure exists outside the OpenShift cluster. It enables integration with enterprise-grade storage solutions, such as IBM Storage Ceph, IBM FlashSystems, IBM Storage Scale, or IBM Fusion Access for Storage Area Network (SAN), that reside outside the OpenShift environment. For instructions, see Configuring Fusion Data Foundation in external mode.

Provider
: In provider mode, you can run a local Fusion Data Foundation storage cluster that serves as the central storage provider for multiple consumer clusters. It functions similarly to an external Fusion Data Foundation configuration, where the provider cluster acts as the provider and base storage on the host cluster. You can provision new clusters within a managed rack and have them consume storage from the central provider cluster. This mode is available from IBM Fusion 2.8.1 onward with Fusion Data Foundation 4.16. For instructions, see Configuring Fusion Data Foundation in provider mode.

  In this mode, Fusion Data Foundation is layered on Red Hat OpenShift Service on bare metal and includes the Fusion Data Foundation provider. IBM Fusion with Fusion Data Foundation supplies the base storage for all Hosted Control Plane-based clusters, as shown in the following figure:

  Figure 1. Fusion Data Foundation provider mode



  Fusion Data Foundation provider mode is a deployment configuration in which the OpenShift cluster running on bare metal serves local storage for OpenShift virtualization, standard applications, and hosted clusters that are running on top of OpenShift virtualization. Provider mode consists of two key components:

  - **Provider** - The provider is a deployment mode for Fusion Data Foundation storage services, built on a minimum of three worker nodes using bare metal volumes. This mode allows external clusters to connect through the client operator. Fusion Data Foundation replicates data across these nodes, ensuring resilience and tolerance to node failures.
  - **Client** - The client is a managed Container Storage Interface (CSI) deployment that connects to the provider to deliver a local storage class for hosted clusters.

  From FDF 4.19 onward, provider and internal deployments use the same unified configuration. This you can deploy a central OpenShift cluster that provides storage for local workloads, OpenShift virtualization clusters, hosted clusters, and standard clusters. If you plan to use provider mode in the future, ensure that Multus is not enabled. After deployment, you can run the provider cluster locally and add remote clients at a later stage, regardless of where they are hosted.

For instructions on Fusion Data Foundation installation, see Installing Fusion Data Foundation.

After completing the configuration, you can monitor capacity utilization and storage nodes, add nodes, and scale capacity as needed.

- **Configuring Fusion Data Foundation in local mode**
  Configure Fusion Data Foundation in local mode with local storage devices.
- **Configuring Fusion Data Foundation in dynamic mode**
  Configure Fusion Data Foundation in dynamic mode with dynamic storage devices.
- **Configuring Fusion Data Foundation in external mode**
  Configure Fusion Data Foundation in external mode to use IBM Storage Ceph, IBM FlashSystem, IBM Storage Scale, or IBM Fusion Access for Storage Area Network (SAN).
- **Configuring Fusion Data Foundation in provider mode**
  Configure Fusion Data Foundation in provider mode that acts as the provider and base storage for the host cluster.
- **Configuring Fusion Data Foundation in MCG-only mode**
  Configure Fusion Data Foundation in Multicloud Object Gateway (MCG)-only mode to provide scalable object storage without block or file services.

- **Scaling out storage capacity**
  Fusion Data Foundation is highly scalable and can be expanded by adding new nodes with sufficient storage, CPU, and RAM resources.
- **Scaling up storage capacity**
  Add a new disk to a storage node to increase the storage capacity of the Fusion Data Foundation cluster.
- **Enabling automatic capacity scaling**
  Enable automatic capacity scaling on the Fusion Data Foundation cluster configured with dynamic storage devices to ensure seamless scaling to meet demand.

Configuring Fusion Data Foundation in local mode

Configure Fusion Data Foundation in local mode with local storage devices.

Before you begin

- Ensure that the disk type is SSD, as it is the only supported disk type.
- If you plan to use the multi-network plug-in (Multus), create the network attachment definitions (NADs) before deployment. These NADs are later attached to the cluster. For more information, see Multi network plug-in (Multus) support and Creating network attachment definitions.
- Fusion Data Foundation service is installed with device type set to Local, as instructed in Installing Fusion Data Foundation. For more information about local storage devices, see Local storage devices.
- Do not use GPU nodes to configure Fusion Data Foundation, as these nodes are dedicated for AI workloads.
- If Fusion Data Foundation is deployed on IBM Z, see Managing disks in a KVM environment in the *Red Hat OpenShift Data Foundation documentation* to manage disks for KVM guests, including adding disks and configuring device properties.

Procedure

1. On the Local storage page of the IBM Fusion web console, click Get started to initiate configuration of the Fusion Data Foundation service.
   It redirects you to the Create StorageSystem page in the Red Hat® OpenShift® Container Platform web console.
   Note: Ensure that you are logged in to the Red Hat OpenShift Container Platform web console as an administrator.
2. In the Backing storage page, select the Create a new StorageClass using the local storage devices option and click Next.
3. In the Advanced Settings page, you can configure the following optional items:
   - Enable Network Files System

     Automatically enable NFS for the cluster

   - Use Ceph RBD as the default StorageClass

     Sets Ceph RBD as the default StorageClass, eliminating the need to manually annotate a StorageClass.

   - Set default StorageClass for virtualization

     Marks the RBD virtualization storage class as the default for KubeVirt VM disks (PVs) after installation.

   - Use external PostgreSQL [Technology preview]

     Enables the use of an external PostgreSQL instance. This provides a high-availability solution for the Multicloud Object Gateway, where the PostgreSQL pod is a single point of failure.

     Important:
     Fusion Data Foundation ships PostgreSQL images maintained by IBM, which are used to store metadata for the Multicloud Object Gateway. This PostgreSQL usage is at the application level.

     As a result, Fusion Data Foundation does not perform database-level optimizations or in-depth insights.

     If you have well-maintained and optimized PostgreSQL instance, then it is recommended to use it. Fusion Data Foundation supports external PostgreSQL instances.

     Any PostgreSQL-related issues requiring code changes or deep technical analysis may need to be addressed upstream. This could result in longer resolution times.

     - Provide the following connection details:
       - Username
       - Password
       - Server name and Port
       - Database name
     - Select Enable TLS/SSL checkbox to enable encryption for the Postgres server.
   - Enable automatic backup

     Allows automatic backup for the Multicloud Object Gateway metadata database.

     - Select the backup frequency: Daily, Weekly, or Monthly.
     - Specify the number of backups to retain.

       Note: This option is disabled if the Use external PostgreSQL option is selected.
     - Click Next.
4. On the Create local volume set tab, do as follows:
   a. Enter a name for the LocalVolumeSet and the StorageClass.
      The local volume set name appears as the default value for the storage class name. You can change the name.
      Remember: The storage class name must not exceed 40 characters and must not match any of the following predefined storage classes:
      - `ocs-storagecluster-cephfs`
      - `ocs-storagecluster-ceph-rbd`

- `ocs-storagecluster-ceph-rgw`
- `openshift-storage.noobaa.io`
- `ocs-storagecluster-ceph-rbd-virtualization`

b. Select one of the following options:

- Disks on all nodes

  Uses the available disks that match the selected filters on all the nodes.

- Disks on selected nodes

  Uses the available disks that match the selected filters only on the selected nodes.

Important:

- The flexible scaling feature is enabled only when the storage cluster that you created with three or more nodes are spread across fewer than the minimum requirement of three availability zones.

  For information about flexible scaling, see knowledgebase article on *Scaling OpenShift Data Foundation cluster using YAML when flexible scaling is enabled*.

- Flexible scaling features get enabled at the time of deployment and cannot be enabled or disabled later on.
- If the nodes selected do not match the Fusion Data Foundation cluster requirement of an aggregated 30 CPUs and 72 GiB of RAM, a minimal cluster is deployed.

  For minimum starting node requirements, see Resource requirements.

c. From the available list of Disk Type, select **`SSD/NVMe`**.
Important: Ensure that you cleaned the disk before selecting it. For instructions, see Red Hat knowledge base article Uninstalling OpenShift Data Foundation in Internal mode.

d. Expand the Advanced section and set the following options:

- Volume Mode: Block is selected as the default value.
- Device Type: Select one or more device types from the drop-down list
- Disk Size: Set a minimum size of 100 GB for the device and maximum available size of the device that needs to be included.
- Maximum Disks Limit: This indicates the maximum number of Persistent Volumes (PVs) that you can create on a node. If this field is left empty, then PVs are created for all the available disks on the matching nodes

e. Click Next.
A pop-up appears to confirm the creation of the LocalVolumeSet.

f. Click Yes to continue.

5. On the Capacity and nodes tab, do as follows:

a. Check the Available raw capacity.
Available raw capacity is populated with the capacity value based on all the attached disks that are associated with the storage class. This takes some time to show up. The Selected nodes list shows the nodes based on the storage class.

b. In the Configure performance tab, select one of the following resource profiles:
Important: Before selecting a resource profile, ensure to check the current availability of resources within the cluster. Opting for a higher resource profile in a cluster with insufficient resources might lead to installation failures.

- Lean

  Use this in a resource constrained environment with minimum resources that are lower than the recommended. This profile minimizes resource consumption by allocating fewer CPUs and less memory.

- Balanced (default)

  Use this when recommended resources are available. This profile provides a balance between resource consumption and performance for diverse workloads.

- Performance

  Use this in an environment with sufficient resources to get the best performance. This profile is tailored for high performance by allocating ample memory and CPUs to ensure optimal execution of demanding workloads.

For more information about resource requirements, see Resource requirements for performance profiles.

Note: You can configure the resource profile even after the deployment by selecting the Configure performance from the options menu in the StorageSystems tab.

c. Optional: Select the Taint nodes checkbox to dedicate the selected nodes for Fusion Data Foundation.

d. Click Next.

6. Optional: On the Security and network tab, configure the following based on your requirement:

a. Select Enable data encryption for block and file storage to enable encryption.

b. Select one or both of the following Encryption level:

- **Cluster-wide encryption**

  Encrypts the entire cluster (block and file).

- **StorageClass encryption**

  Creates encrypted persistent volume (block only) using encryption-enabled storage class.

c. Optional: Select the Connect to an external key management service checkbox.
This is optional for cluster-wide encryption.

i. From the Key Management Service Provider drop-down list, either select Vault or Thales CipherTrust Manager (using KMIP).

If you selected Vault, do to the next step. If you selected Thales CipherTrust Manager (using KMIP), do step 6.c.iii.

ii. Select either of the following Authentication Method:

- **Using Token authentication method**
  - Enter a unique Connection Name, host Address of the Vault server ('https://<hostname or ip>'), Port number, and Token.
  - Expand Advanced Settings to enter additional settings and certificate details based on your `Vault` configuration:
    - Enter the Key Value secret path in Backend Path that is dedicated and unique to Fusion Data Foundation.
    - Optional: Enter TLS Server Name and Vault Enterprise Namespace.
    - Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate, and Client Private Key.
  - Click Save and skip to step 6.d.
- **Using Kubernetes authentication method**
  - Enter a unique Connection Name, host Address of the Vault server ('https://<hostname or ip>'), Port number, and Role name.
  - Expand Advanced Settings to enter additional settings and certificate details based on your `Vault` configuration:
    - Enter the Key Value secret path in Backend Path that is dedicated and unique to Fusion Data Foundation.
    - Optional: Enter TLS Server Name and Vault Enterprise Namespace.
    - Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate, and Client Private Key.
  - Click Save and skip to step 6.d.

    Note: In case you need to enable key rotation for Vault KMS, run the following command in the Red Hat OpenShift Container Platform web console after the storage cluster is created:

    ```
    oc patch storagecluster ocs-storagecluster -n openshift-storage --type=json -p '[{"op": "add", "path":"/spec/encryption/keyRotation/enable", "value": true}]'
    ```

iii. To use **Thales CipherTrust Manager (using KMIP)** as the KMS provider, do the following steps:

1. Enter a unique Connection Name for the Key Management service within the project.
2. In the Address and Port sections, enter the IP of Thales CipherTrust Manager and the port where the KMIP interface is enabled. For example:
   - Address: 123.34.3.2
   - Port: 5696
3. Upload the Client Certificate, CA certificate, and Client Private Key.
4. If StorageClass encryption is enabled, enter the Unique Identifier to be used for encryption and decryption that is generated in the previous step.
5. The TLS Server field is optional and used when there is no DNS entry for the KMIP endpoint. For example, `kmip_all_<port>.ciphertrustmanager.local`.

d. Select one of the following Network options:

- Default (OVN)

  If you are using a single network.

- Custom (Multus)

  If you are using multiple network interfaces.

  - Select a Public Network Interface from the drop-down.
  - Select a Cluster Network Interface from the drop-down.

    Note: If you are using only one additional network interface, select the single `NetworkAttachementDefinition`, that is, `ocs-public-cluster` for the Public Network Interface and leave the Cluster Network Interface blank.

e. Click Next.

7. On the Data protection tab, select Prepare cluster for disaster recovery (Regional-DR only) and click Next.
8. On the Review and create tab, review the configuration details.
   To modify any configuration settings, click Back to go back to the previous configuration page.
9. Click Create StorageSystem.
   Note: When your deployment has five or more nodes, racks, or rooms, and when there are five or more number of failure domains present in the deployment, you can configure Ceph monitor counts based on the number of racks or zones. An alert is displayed in the notification panel or Alert Center of the Red Hat OpenShift Container Platform web console to indicate the option to increase the number of Ceph monitor counts. You can use the Configure option in the alert to configure the Ceph monitor counts. For more information, see Resolving low Ceph monitor count alert.
10. Validate whether Fusion Data Foundation is successfully configured in local mode by following the instructions that are mentioned in Verifying Fusion Data Foundation local mode configuration.

What to do next

- To add new nodes to increase storage capacity in the cluster, see Scaling out storage capacity.
- To add new disks to a storage node, see Scaling up storage capacity.
- To configure Kubernetes authentication method, see Enable encryption with KMS using the Kubernetes authentication method.

- **Verifying Fusion Data Foundation local mode configuration**
  Validate whether Fusion Data Foundation is successfully configured in local mode.

Verifying Fusion Data Foundation local mode configuration

Validate whether Fusion Data Foundation is successfully configured in local mode.

Procedure

1. Verify the final status of the installed storage cluster.
   a. Go to Storage > Data Foundation > Storage System > ocs-storagecluster.
   b. Verify that the `Status` of the `StorageCluster` is `Ready` and has a green tick mark next to it.
2. Verify whether the flexible scaling is enabled on your storage cluster (for arbiter mode, flexible scaling is disabled).

a. Go to Storage > Data Foundation > Storage System > ocs-storagecluster.
b. From the Action menu, select Edit Storage System.
c. In the YAML tab, search for the keys **flexibleScaling** in the **spec** section and **failureDomain** in the **status** section.
If **flexible scaling** is true and **failureDomain** is set to host, flexible scaling feature is enabled.

```
spec:
flexibleScaling: true
[…]
status:
failureDomain: host
```

3. Verify the state of the Pods by following the instructions that are mentioned in Verifying the state of the Pods.
4. Verify that the Fusion Data Foundation cluster is healthy by following the instructions that are mentioned in Verifying the Fusion Data Foundation cluster is healthy.
5. Verify that the MCG is healthy by following the instructions that are mentioned in Verifying the Multicloud Object Gateway is healthy.
6. Verify that the Fusion Data Foundation specific storage classes exist by following the instructions that are mentioned in Verifying that the specific storage classes exist.
7. Verify the Multus networking by following the instructions that are mentioned in Verifying the Multus networking.

- **Verifying the state of the Pods**
  Validate the state of Pods to ensure that all components are running correctly and the Fusion Data Foundation cluster is healthy.
- **Verifying the Fusion Data Foundation cluster is healthy**
  Check the health of the Fusion Data Foundation cluster to ensure Fusion Data Foundation is operating as expected.
- **Verifying the Multicloud Object Gateway is healthy**
  Check the health status of the Multicloud Object Gateway (MCG) to confirm its availability in the Fusion Data Foundation cluster.
- **Verifying that the specific storage classes exist**
  Confirm that necessary storage classes are available in the Fusion Data Foundation cluster to support workloads and maintain consistent storage operations.
- **Verifying the Multus networking**
  Verify that the Multus networking in configured in your Fusion Data Foundation cluster.

Verifying the state of the Pods

Validate the state of Pods to ensure that all components are running correctly and the Fusion Data Foundation cluster is healthy.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, select openshift-storage from the Project drop-down list.
   Note: If the Show default projects option is disabled, use the toggle button to list all the default projects.
2. Go to Workloads > Pods.
   Set filter for running and completed pods to verify that the following pods are in **Running** and **Completed** state:

   Note: The available Pods depend on the cluster configuration. When the cluster is configured in Multicloud Object Gateway (MCG) mode, the **rook-ceph-operator-*** Pods are not available. Similarly, when the cluster is deployed without the MCG, **noobaa-*** Pods are not available.

Table 1. Pods corresponding to Fusion Data Foundation

Component	Pods
Fusion Data Foundation Operator	• **ocs-operator-*** (1 pod on any storage node) • **ocs-metrics-exporter-*** (1 pod on any storage node) • **odf-operator-controller-manager-*** (1 pod on any storage node) • **odf-console-*** (1 pod on any storage node) • **csi-addons-controller-manager-*** (1 pod on any storage node)
Rook-ceph Operator	**rook-ceph-operator-*** (1 pod on any storage node)
Multicloud Object Gateway	• **noobaa-operator-*** (1 pod on any storage node) • **noobaa-core-*** (1 pod on any storage node) • **noobaa-db-pg-cluster-1*** and **noobaa-db-pg-cluster-2*** (2 instances of MCG DB pod on any storage node) • **noobaa-endpoint-*** (1 pod on any storage node) • **cnpg-controller-manager-*** (1 pod on any storage node)
MON	**rook-ceph-mon-*** (3 pods distributed across storage nodes)
MGR	**rook-ceph-mgr-*** (2 pods on different storage nodes, one active, one standby)
MDS	**rook-ceph-mds-ocs-storagecluster-cephfilesystem-*** (2 pods distributed across storage nodes)
RGW	**rook-ceph-rgw-ocs-storagecluster-cephobjectstore-*** (1 pod on any storage node)

CSI	• **cephfs**
	  • **openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*** (2 pods distributed across storage nodes)
	  • **openshift-storage.cephfs.csi.ceph.com-nodeplugin-*** (1 pod on each storage node)
	• **nfs**
	  • **openshift-storage.nfs.csi.ceph.com-ctrlplugin-*** (2 pods distributed across storage nodes)
	  • **openshift-storage.nfs.csi.ceph.com-nodeplugin-*** (1 pod on each storage node)
	• **rbd**
	  • **openshift-storage.rbd.csi.ceph.com-ctrlplugin-*** (2 pods distributed across storage nodes)
	  • **openshift-storage.rbd.csi.ceph.com-nodeplugin-*** (1 pod on each storage node)

rook-ceph-crashcollector	**rook-ceph-crashcollector-*** (1 pod on each storage node)
OSD	**rook-ceph-osd-*** (1 pod for each device)

Verifying the Fusion Data Foundation cluster is healthy

Check the health of the Fusion Data Foundation cluster to ensure Fusion Data Foundation is operating as expected.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. In the Status card of the Block and File tab, verify that the Storage Cluster has a green tick.
4. In the Details card, verify that the cluster information is displayed.
   For more information on the health of the Fusion Data Foundation cluster using the Block and File dashboard, see Monitoring Fusion Data Foundation.

Verifying the Multicloud Object Gateway is healthy

Check the health status of the Multicloud Object Gateway (MCG) to confirm its availability in the Fusion Data Foundation cluster.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. In the Status card of the Object tab, verify that both Object Service and Data Resiliency have a green tick.
4. In the Details card, verify that the MCG information is displayed.
   For more information on the health of the Fusion Data Foundation cluster using the object service dashboard, see Monitoring Fusion Data Foundation.

   Important: To avoid data loss, it is recommended to take a backup of NooBaa DB PVC regularly. If NooBaa DB fails and cannot be recovered, then you can revert to the latest backed-up version. For instructions on backing up your NooBaa DB, follow the steps in the knowledgebase article, Perform a One-Time Backup of the Database for the Multicloud Object Gateway.

Verifying that the specific storage classes exist

Confirm that necessary storage classes are available in the Fusion Data Foundation cluster to support workloads and maintain consistent storage operations.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Storage Classes.
2. Verify that the following storage classes are created with the Fusion Data Foundation cluster creation:
   - `ocs-storagecluster-ceph-rbd`
   - `ocs-storagecluster-ceph-rgw`
   - `ocs-storagecluster-cephfs`
   - `openshift-storage.noobaa.io`

Verifying the Multus networking

Verify that the Multus networking in configured in your Fusion Data Foundation cluster.

About this task

Based on your network configuration choices, the Fusion Data Foundation operator does one of the following:

- If only a single NetworkAttachmentDefinition (for example, `ocs-public-cluster`) was selected for the Public Network Interface, then the traffic between the application pods and the Fusion Data Foundation cluster happens on this network. Additionally the cluster is self configured to also use this network for the replication and rebalancing traffic between OSDs.
- If both NetworkAttachmentDefinitions (for example, `ocs-public` and `ocs-cluster`) were selected for the Public Network Interface and the Cluster Network Interface respectively during the Storage Cluster installation, then client storage traffic is on the public network and cluster network for the replication and rebalancing traffic between OSDs.

To verify the network configuration is correct, complete the following:

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Data Foundation > Storage System.
2. From the Action menu, select Edit Storage System.
3. In the YAML tab, search for `network` in the `spec` section and ensure the configuration is correct for your network interface choices.
   This example is for separating the client storage traffic from the storage replication traffic.

   Sample output:

```
[..]
spec:
  [..]
  network:
    ipFamily: IPv4
```

```
    provider: multus
    selectors:
      cluster: openshift-storage/ocs-cluster
      public: openshift-storage/ocs-public
  [..]
```

To verify the network configuration is correct using the command line interface, run the following commands:

```
oc get storagecluster ocs-storagecluster \
-n openshift-storage \
-o=jsonpath='{.spec.network}{"\n"}'
```

Sample output:

```
{"ipFamily":"IPv4","provider":"multus","selectors":{"cluster":"openshift-storage/ocs-cluster","public":"openshift-storage/ocs-
```

4. Confirm the OSD pods are using correct network.
   In the **openshift-storage** namespace use one of the OSD pods to verify the pod has connectivity to the correct networks.

   Note: Only the OSD pods connect to both Multus public and cluster networks if both are created. All other OCS pods connect to the Multus public network. This example is for separating the client storage traffic from the storage replication traffic.

   ```
   oc get -n openshift-storage $(oc get pods -n openshift-storage -o name -l app=rook-ceph-osd | grep 'osd-0') -
   o=jsonpath='{.metadata.annotations.k8s\.v1\.cni\.cncf\.io/network-status}{"\n"}'
   ```

   Sample output:

   ```
   [{
       "name": "openshift-sdn",
       "interface": "eth0",
       "ips": [
           "10.129.2.30"
       ],
       "default": true,
       "dns": {}
   },{
       "name": "openshift-storage/ocs-cluster",
       "interface": "net1",
       "ips": [
           "192.168.2.1"
       ],
       "mac": "e2:04:c6:81:52:f1",
       "dns": {}
   },{
       "name": "openshift-storage/ocs-public",
       "interface": "net2",
       "ips": [
           "192.168.1.1"
       ],
       "mac": "ee:a0:b6:a4:07:94",
       "dns": {}
   }]
   ```

   To confirm the OSD pods are using correct network using the command line interface, run the following command (requires the jq utility):

   ```
   oc get -n openshift-storage $(oc get pods -n openshift-storage -o name -l app=rook-ceph-osd | grep 'osd-0') -
   o=jsonpath='{.metadata.annotations.k8s\.v1\.cni\.cncf\.io/network-status}{"\n"}' | jq -r '.[].name'
   ```

   Sample output:

   ```
   openshift-sdn
   openshift-storage/ocs-cluster
   openshift-storage/ocs-public
   ```

Configuring Fusion Data Foundation in dynamic mode

Configure Fusion Data Foundation in dynamic mode with dynamic storage devices.

Before you begin

- Fusion Data Foundation service is installed with device type set to Dynamic, as instructed in Installing Fusion Data Foundation. For more information about local storage devices, see Dynamic storage devices.
- Do not use GPU nodes to configure Fusion Data Foundation, as these nodes are dedicated for AI workloads.
- Applicable for Microsoft Azure: If you intend to use StorageClass with performance plus enabled, you must create a StorageClass with **enablePerformancePlus=True** and select this StorageClass during deployment.

Procedure

1. On the Local storage page of the IBM Fusion web console, click Get started to initiate configuration of the Fusion Data Foundation service.
   It redirects you to the Create StorageSystem page in the Red Hat® OpenShift® Container Platform web console.
   Note: Ensure that you are logged in to the Red Hat OpenShift Container Platform web console as an administrator.
2. In the Backing storage page, select the following:
   a. Select the Use an existing StorageClass option.
   b. Select the Storage Class.
      You can choose **gp2-csi** or **gp3-csi** as the storage class.

c. Click Next.

3. In the Advanced Settings page, you can configure the following optional items:
   - Enable Network Files System

     Automatically enable NFS for the cluster

   - Use Ceph RBD as the default StorageClass

     Sets Ceph RBD as the default StorageClass, eliminating the need to manually annotate a StorageClass.

   - Set default StorageClass for virtualization

     Marks the RBD virtualization storage class as the default for KubeVirt VM disks (PVs) after installation.

   - Use external PostgreSQL [Technology preview]

     Enables the use of an external PostgreSQL instance. This provides a high-availability solution for the Multicloud Object Gateway, where the PostgreSQL pod is a single point of failure.

     Important:
     Fusion Data Foundation ships PostgreSQL images maintained by IBM, which are used to store metadata for the Multicloud Object Gateway. This PostgreSQL usage is at the application level.

     As a result, Fusion Data Foundation does not perform database-level optimizations or in-depth insights.

     If you have well-maintained and optimized PostgreSQL instance, then it is recommended to use it. Fusion Data Foundation supports external PostgreSQL instances.

     Any PostgreSQL-related issues requiring code changes or deep technical analysis may need to be addressed upstream. This could result in longer resolution times.

     - Provide the following connection details:
       - Username
       - Password
       - Server name and Port
       - Database name
     - Select Enable TLS/SSL checkbox to enable encryption for the Postgres server.
   - Enable automatic backup

     Allows automatic backup for the Multicloud Object Gateway metadata database.

     - Select the backup frequency: Daily, Weekly, or Monthly.
     - Specify the number of backups to retain.

       Note: This option is disabled if the Use external PostgreSQL option is selected.
     - Click Next.

4. On the Capacity and nodes tab, do as follows:
   a. Select a value for Requested Capacity from the drop-down list.
   By default, `2 TiB` is set.
   Note: After you select the initial storage capacity, cluster expansion is performed only by using the selected usable capacity (three times the raw storage).
   b. In the Select Nodes tab, select at least three available nodes.
   c. In the Configure performance tab, select one of the following resource profiles:
   Important: Before selecting a resource profile, ensure to check the current availability of resources within the cluster. Opting for a higher resource profile in a cluster with insufficient resources might lead to installation failures.
      - Lean

        Use this in a resource constrained environment with minimum resources that are lower than the recommended. This profile minimizes resource consumption by allocating fewer CPUs and less memory.

      - Balanced (default)

        Use this when recommended resources are available. This profile provides a balance between resource consumption and performance for diverse workloads.

      - Performance

        Use this in an environment with sufficient resources to get the best performance. This profile is tailored for high performance by allocating ample memory and CPUs to ensure optimal execution of demanding workloads.

     For more information about resource requirements, see Resource requirements for performance profiles.

     Note: You can configure the resource profile even after the deployment by selecting the Configure performance from the options menu in the StorageSystems tab.
   d. Optional: Select the Taint nodes checkbox to dedicate the selected nodes for Fusion Data Foundation.
   For cloud platforms with multiple availability zones, ensure that the nodes are distributed across different locations or availability zones.

     If the nodes selected do not match the Fusion Data Foundation cluster requirement of an aggregated 30 CPUs and 72 GiB of RAM, a minimal cluster is deployed. For minimum starting node requirements, see Resource requirements.

   e. Optional: Select the Enable automatic capacity scaling for your cluster checkbox.
   When automatic capacity scaling is enabled, additional raw capacity equivalent to the configured deployment size is automatically added to the cluster when the used capacity reaches 70%. This ensures your Fusion Data Foundation cluster scales seamlessly to meet demand.

     This option is disabled in lean profile mode, local mode, and external mode deployment.

     Important: Enabling automatic capacity scaling may incur additional costs for the underlying storage.

Set the cluster expansion limit from the drop-down list. This defines the maximum capacity that the cluster can expand to in the cloud. Automatic scaling is suspended once this limit is reached.

f. Click Next.

5. Optional: On the Security and network tab, configure the following based on your requirement:
   a. Select Enable data encryption for block and file storage to enable encryption.
   b. Select one or both of the following Encryption level:
      - **Cluster-wide encryption**

        Encrypts the entire cluster (block and file).

      - **StorageClass encryption**

        Creates encrypted persistent volume (block only) using encryption enabled storage class.

   c. Optional: Select the Connect to an external key management service checkbox.
      This is optional for cluster-wide encryption.

      i. From the Key Management Service Provider drop-down list, either select Vault or Thales CipherTrust Manager (using KMIP).

      If you selected Vault, do to the next step. If you selected Thales CipherTrust Manager (using KMIP), do step 5.c.iii.

      ii. Select either of the following Authentication Method:

      - **Using Token authentication method**
        - Enter a unique Connection Name, host Address of the Vault server ('https://<hostname or ip>'), Port number, and Token.
        - Expand Advanced Settings to enter additional settings and certificate details based on your `Vault` configuration:
          - Enter the Key Value secret path in Backend Path that is dedicated and unique to Fusion Data Foundation.
          - Optional: Enter TLS Server Name and Vault Enterprise Namespace.
          - Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate, and Client Private Key.
        - Click Save and skip to step 5.c.iv.
      - **Using Kubernetes authentication method**
        - Enter a unique Connection Name, host Address of the Vault server ('https://<hostname or ip>'), Port number, and Role name.
        - Expand Advanced Settings to enter additional settings and certificate details based on your `Vault` configuration:
          - Enter the Key Value secret path in Backend Path that is dedicated and unique to Fusion Data Foundation.
          - Optional: Enter TLS Server Name, Authentication Path, and Vault Enterprise Namespace.
          - Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate, and Client Private Key.
        - Click Save and skip to step 5.c.iv.
          Note: To enable key rotation for Vault KMS, run the following command:

          ```
          oc patch storagecluster ocs-storagecluster -n openshift-storage --type=json -p '[{"op": "add", "path":"/spec/encryption/keyRotation/enable", "value": true}]'
          ```

      iii. To use **Thales CipherTrust Manager (using KMIP)** as the KMS provider, do the following steps:

      1. Enter a unique Connection Name for the Key Management service within the project.
      2. In the Address and Port sections, enter the IP of Thales CipherTrust Manager and the port where the KMIP interface is enabled. For example:
         - Address: 123.34.3.2
         - Port: 5696
      3. Upload the Client Certificate, CA certificate, and Client Private Key.
      4. If StorageClass encryption is enabled, enter the Unique Identifier to be used for encryption and decryption generated in the previous step.
      5. The TLS Server field is optional and used when there is no DNS entry for the KMIP endpoint. For example, `kmip_all_<port>.ciphertrustmanager.local`.

      iv. Select In-transit encryption to enable in-transit encryption.
      1. Select a Network.
      2. Click Next.

6. On the Review and create tab, review the configuration details.
   To modify any configuration settings, click Back to go back to the previous configuration page.
7. Click Create StorageSystem.
   Note: When your deployment has five or more nodes, racks, or rooms, and when there are five or more number of failure domains present in the deployment, you can configure Ceph monitor counts based on the number of racks or zones. An alert is displayed in the notification panel or Alert Center of the Red Hat OpenShift Container Platform web console to indicate the option to increase the number of Ceph monitor counts. You can use the Configure option in the alert to configure the Ceph monitor counts. For more information, see Resolving low Ceph monitor count alert.
8. Validate whether Fusion Data Foundation is successfully configured in dynamic mode by following the instructions that are mentioned in Verifying Fusion Data Foundation dynamic mode configuration.

What to do next

- To add new nodes to increase storage capacity in the cluster, see Scaling out storage capacity.
- To add new disks to a storage node, see Scaling up storage capacity.
- To configure Kubernetes authentication method, see Enable encryption with KMS using the Kubernetes authentication method.

- **Verifying Fusion Data Foundation dynamic mode configuration**
  Validate whether Fusion Data Foundation is successfully configured in dynamic mode.

Verifying Fusion Data Foundation dynamic mode configuration

Validate whether Fusion Data Foundation is successfully configured in dynamic mode.

Procedure

1. Verify the final status of the installed storage cluster.
    a. Go to Storage > Data Foundation > Storage System > ocs-storagecluster.
    b. Verify that the **Status** of the **StorageCluster** is **Ready** and has a green tick mark next to it.
2. Verify the state of the Pods by following the instructions that are mentioned in Verifying the state of the Pods.
3. Verify that the Fusion Data Foundation cluster is healthy by following the instructions that are mentioned in Verifying the Fusion Data Foundation cluster is healthy.
4. Verify that the MCG is healthy by following the instructions that are mentioned in Verifying the Multicloud Object Gateway is healthy.
5. Verify that the Fusion Data Foundation specific storage classes exist by following the instructions that are mentioned in Verifying that the specific storage classes exist.

- **Verifying the state of the Pods**
  Validate the state of Pods to ensure that all components are running correctly and the Fusion Data Foundation cluster is healthy.
- **Verifying the Fusion Data Foundation cluster is healthy**
  Check the health of the Fusion Data Foundation cluster to ensure Fusion Data Foundation is operating as expected.
- **Verifying the Multicloud Object Gateway is healthy**
  Check the health status of the Multicloud Object Gateway (MCG) to confirm its availability in the Fusion Data Foundation cluster.
- **Verifying that the specific storage classes exist**
  Confirm that necessary storage classes are available in the Fusion Data Foundation cluster to support workloads and maintain consistent storage operations.

Verifying the state of the Pods

Validate the state of Pods to ensure that all components are running correctly and the Fusion Data Foundation cluster is healthy.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, select openshift-storage from the Project drop-down list.
   Note: If the Show default projects option is disabled, use the toggle button to list all the default projects.
2. Go to Workloads > Pods.
   Set filter for running and completed pods to verify that the following pods are in **Running** and **Completed** state:

   Note: The available Pods depend on the cluster configuration. When the cluster is configured in Multicloud Object Gateway (MCG) mode, the **rook-ceph-operator-*** Pods are not available. Similarly, when the cluster is deployed without the MCG, **noobaa-*** Pods are not available.

Table 1. Pods corresponding to Fusion Data Foundation

Component	Pods
Fusion Data Foundation Operator	• **ocs-operator-*** (1 pod on any storage node) • **ocs-metrics-exporter-*** (1 pod on any storage node) • **odf-operator-controller-manager-*** (1 pod on any storage node) • **odf-console-*** (1 pod on any storage node) • **csi-addons-controller-manager-*** (1 pod on any storage node) • **ux-backend-server-*** (1 pod on any storage node) • **ocs-client-operator-*** (1 pod on any storage node) • **ocs-client-operator-console-*** (1 pod on any storage node) • **ocs-provider-server-*** (1 pod on any storage node)
Rook-ceph Operator	**rook-ceph-operator-*** (1 pod on any storage node)
Multicloud Object Gateway	• **noobaa-operator-*** (1 pod on any storage node) • **noobaa-core-*** (1 pod on any storage node) • **noobaa-db-pg-cluster-1*** and **noobaa-db-pg-cluster-2*** (2 instances of MCG DB pod on any storage node) • **noobaa-endpoint-*** (1 pod on any storage node) • **cnpg-controller-manager-*** (1 pod on any storage node)
MON	**rook-ceph-mon-*** (3 pods distributed across storage nodes)
MGR	**rook-ceph-mgr-*** (2 pods on different storage nodes, one active, one standby)
MDS	**rook-ceph-mds-ocs-storagecluster-cephfilesystem-*** (2 pods distributed across storage nodes)

CSI	• **cephfs**
	  • **openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*** (2 pods distributed across storage nodes)
	  • **openshift-storage.cephfs.csi.ceph.com-nodeplugin-*** (1 pod on each storage node)
	• **nfs**
	  • **openshift-storage.nfs.csi.ceph.com-ctrlplugin-*** (2 pods distributed across storage nodes)
	  • **openshift-storage.nfs.csi.ceph.com-nodeplugin-*** (1 pod on each storage node)
	• **rbd**
	  • **openshift-storage.rbd.csi.ceph.com-ctrlplugin-*** (2 pods distributed across storage nodes)
	  • **openshift-storage.rbd.csi.ceph.com-nodeplugin-*** (1 pod on each storage node)

rook-ceph-crashcollector	**rook-ceph-crashcollector-*** (1 pod on each storage node)
OSD	• **rook-ceph-osd-*** (1 pod for each device) • **rook-ceph-osd-prepare-ocs-*** (1 pod for each device)
ceph-csi-operator	**ceph-csi-controller-manager-*** (1 pod for each device)
odf-blackbox-exporter	**odf-blackbox-exporter-*** (1 pod on any storage node) Note: This does not apply to clusters without storage, for example, Multicloud Object Gateway-only deployments or external mode deployments.

Verifying the Fusion Data Foundation cluster is healthy

Check the health of the Fusion Data Foundation cluster to ensure Fusion Data Foundation is operating as expected.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. In the Status card of the Block and File tab, verify that the Storage Cluster has a green tick.
4. In the Details card, verify that the cluster information is displayed.
   For more information on the health of the Fusion Data Foundation cluster using the Block and File dashboard, see Monitoring Fusion Data Foundation.

Verifying the Multicloud Object Gateway is healthy

Check the health status of the Multicloud Object Gateway (MCG) to confirm its availability in the Fusion Data Foundation cluster.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. In the Status card of the Object tab, verify that both Object Service and Data Resiliency have a green tick.
4. In the Details card, verify that the MCG information is displayed.
   For more information on the health of the Fusion Data Foundation cluster using the object service dashboard, see Monitoring Fusion Data Foundation.

   Important: To avoid data loss, it is recommended to take a backup of NooBaa DB PVC regularly. If NooBaa DB fails and cannot be recovered, then you can revert to the latest backed-up version. For instructions on backing up your NooBaa DB, follow the steps in the knowledgebase article, Perform a One-Time Backup of the Database for the Multicloud Object Gateway.

Verifying that the specific storage classes exist

Confirm that necessary storage classes are available in the Fusion Data Foundation cluster to support workloads and maintain consistent storage operations.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Storage Classes.
2. Verify that the following storage classes are created with the Fusion Data Foundation cluster creation:
   - `ocs-storagecluster-ceph-rbd`
   - `ocs-storagecluster-cephfs`
   - `openshift-storage.noobaa.io`

Configuring Fusion Data Foundation in external mode

Configure Fusion Data Foundation in external mode to use IBM Storage Ceph, IBM FlashSystem, IBM Storage Scale, or IBM Fusion Access for Storage Area Network (SAN).

You can configure Fusion Data Foundation in external mode by connecting to one of the following external storage providers:

- IBM Storage Ceph
- IBM FlashSystem
- IBM Storage Scale
- IBM Fusion Access for SAN
  Important: In environments running Red Hat® OpenShift® Container Platform 4.21 or later and Fusion Data Foundation 4.21 or later, you can configure IBM Fusion Access for SAN in Fusion Data Foundation as an external system only if Global Data Platform and Content-Aware Storage services are not installed. These services are mutually exclusive. Before planning or deploying IBM Fusion Access for SAN, ensure that your environment does not include Global Data Platform or Content-Aware Storage services.

Note: Do not use GPU nodes to configure Fusion Data Foundation, as these nodes are dedicated for AI workloads.

- **Configuring with IBM Storage Ceph**
  Configure Fusion Data Foundation in external mode with an external IBM Storage Ceph cluster.
- **Configuring with IBM FlashSystem**
  Configure Fusion Data Foundation in external mode with an external IBM FlashSystem cluster.
- **Configuring with IBM Storage Scale**
  Configure IBM Storage Scale (CNSA) to integrate fast, scalable, and reliable file storage with Fusion Data Foundation, ensuring high performance and reliability for enterprise workloads.
- **Configuring with IBM Fusion Access for Storage Area Network**
  Configure IBM Fusion Access for Storage Area Network (SAN) by connecting external SAN storage to Fusion Data Foundation from the Red Hat OpenShift Container Platform web console.

Configuring with IBM Storage Ceph

Configure Fusion Data Foundation in external mode with an external IBM Storage Ceph cluster.

Before you begin

- Fusion Data Foundation service is installed with device type set to External, as instructed in Installing Fusion Data Foundation.
- IBM Storage Ceph must have Ceph Dashboard that is installed and configured. For more information, see Dashboard > Installation and access in the IBM Storage Ceph documentation.
- It is recommended that the external IBM Storage Ceph cluster has the PG Autoscaler enabled.
- The external IBM Storage Ceph cluster must have an existing RBD pool pre-configured for use. If no pool exists, contact your IBM Storage Ceph administrator to create one before proceeding with the Fusion Data Foundation configuration. IBM recommends using a separate pool for each Fusion Data Foundation cluster.
- Optional: If a zonegroup exists apart from the default zonegroup, add the hostname, `rook-ceph-rgw-ocs-external-storagecluster-cephobjectstore.openshift-storage.svc` to the zonegroup, as Fusion Data Foundation sends S3 requests to the RADOS Object Gateways (RGWs) using this hostname.

Procedure

1. Log in to the Red Hat® OpenShift® Container Platform web console as an administrator, and navigate to Storage > External systems.
2. Click Connect external systems.
3. Select Red Hat/IBM Ceph Cluster and click Next.
4. In the Connect Red Hat Ceph storage page, do as follows:
   a. Click the Download script link to download the Python script for extracting Ceph cluster details.
   b. To extract the IBM Storage Ceph cluster details, contact the IBM Storage Ceph administrator to run the downloaded Python script on an IBM Storage Ceph node by using the `admin key`.
      i. Run the following command on the IBM Storage Ceph node to view the list of available arguments:

      ```
      python3 ceph-external-cluster-details-exporter.py --help
      ```

      Important: Use `python` instead of `python3` if the IBM Storage Ceph cluster is deployed on a Red Hat Enterprise Linux 7.x (RHEL 7.x) cluster.
      You can also run the script from inside a MON container (for containerized deployments) or from a MON node (for RPM deployments).

      Note: Use the **yum install cephadm** command, followed by the **cephadm** command to deploy your IBM Storage Ceph cluster using containers. You must pull the IBM Storage Ceph cluster container images by using the `cephadm` command, rather than installing Ceph packages on nodes by using the **yum** command. For more information, see IBM Storage Ceph documentation.
      ii. Retrieve the external cluster details from the IBM Storage Ceph cluster by using one of the following options:

      Configuration file flag
      Use the config-file flag that stores the parameters used during deployment.

      In new deployments, you can save the deployment parameters in a configuration file. This file can also be used during upgrades to preserve existing parameters and add any new parameters. Use the config-file flag to specify the path to the configuration file.

      An example of a configuration saved in the /config.ini file is as follows:

      ```
      [Configurations]
      format = bash
      cephfs-filesystem-name = <filesystem-name>
      rbd-data-pool-name = <pool_name>
      ...
      ```

      Run the following command to set the path to the /config.ini by using the config-file flag:

      ```
      python3 ceph-external-cluster-details-exporter.py --config-file /config.ini
      ```

      Command line flag
      Retrieve the external cluster details from the IBM Storage Ceph cluster and configure the required parameters mentioned in Table 1 for your deployment.

      Command example:

      ```
      python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name ceph-rbd --monitoring-endpoint xxx.xxx.xxx.xxx --monitoring-endpoint-port xxxx --rgw-endpoint xxx.xxx.xxx.xxx:xxxx --run-as-user client.ocs
      ```

Table 1. List of parameters

Parameter type	Parameter name	Description
RBD parameters	rbd-data-pool-name	A mandatory parameter that is used for providing block storage in Fusion Data Foundation.
	rados-namespace	Divides an RBD data pool into separate logical namespaces, used for creating RBD PVC in a `radosNamespace`. Flags required with `rados-namespace` are `restricted-auth-permission` and `k8s-cluster-name`.
	rbd-metadata-ec-pool-name	(Optional) The name of the erasure coded RBD metadata pool.
RGW parameters	rgw-endpoint	(Optional) This parameter is required only if the object storage is to be provisioned through Ceph Rados Gateway for Fusion Data Foundation. Provide the endpoint in the following format: `<ip_address>:<port>` Note: A fully-qualified domain name (FQDN) is also supported in the format `<FQDN>:<PORT>`.
	rgw-pool-prefix	(Optional) The prefix of the RGW pools. If not specified, the default prefix is `default`.

Parameter type	Parameter name	Description
	rgw-tls-cert-path	(Optional) The file path of the RADOS Gateway endpoint TLS certificate. To provide the TLS certificate and RGW endpoint details to the helper script, **ceph-external-cluster-details-exporter.py**, run the following command: **# python3 ceph-external-clustergw-endpoint r-details-exporter.py --rbd-data-pool-name <rbd block pool name> --rgw-endpoint <ip_address>:<port> --rgw-tls-cert-path <file path containing cert>** This creates a resource to create a Ceph Object Store CR such as Kubernetes secret containing the TLS certificate. All the intermediate certificates including private keys need to be stored in the certificate file
	rgw-skip-tls	(Optional) This parameter ignores the TLS certification validation when a self-signed certificate is provided (not recommended).
Monitoring parameters	monitoring-endpoint	(Optional) This parameter accepts comma-separated list of IP addresses of active and standby **mgrs** reachable from the OpenShift Container Platform cluster. If not provided, the value is automatically populated.
	monitoring-endpoint-port	(Optional) It is the port associated with the **ceph-mgr** Prometheus exporter specified by **--monitoring-endpoint**. If not provided, the value is automatically populated.
Ceph parameters	ceph-conf	(Optional) The name of the Ceph configuration file.
	run-as-user	(Optional) This parameter is used for providing name for the Ceph user which is created by the script. If this parameter is not specified, a default user name **client.healthchecker** is created. The permissions for the new user is set as: • caps: [mgr] allow command config • caps: [mon] allow r, allow command quorum_status, allow command version • caps: [osd] allow rwx pool=**RGW_POOL_PREFIX.rgw.meta**, allow r pool=**.rgw.root**, allow rw pool=**RGW_POOL_PREFIX.rgw.control**, allow rx pool=**RGW_POOL_PREFIX.rgw.log**, allow x pool=**RGW_POOL_PREFIX.rgw.buckets.index**
CephFS parameters	cephfs-metadata-pool-name	(Optional) The name of the CephFS metadata pool.
	cephfs-data-pool-name	(Optional) The name of the CephFS data pool.
	cephfs-filesystem-name	(Optional) The name of the CephFS filesystem.
Output parameters	dry-run	(Optional) This parameter helps to print the executed commands without running them.
	output	(Optional) The file where the output is required to be stored.
Multicluster parameters	k8s-cluster-name	(Optional) Kubernetes cluster name.
	cluster-name	(Optional) The Ceph cluster name.
	restricted-auth-permission	(Optional) This parameter restricts **cephCSIKeyrings** auth permissions to specific pools and clusters. Mandatory flags that need to be set with this are **rbd-data-pool-name** and **cluster-name**. You can also pass the **cephfs-filesystem-name** flag if there is CephFS user restriction so that permission is restricted to a particular CephFS filesystem. Note: This parameter must be applied only for the new deployments. To restrict **csi-users**per pool and per cluster, you need to create new **csi-users** and new secrets for those **csi-users**.

Command example with restricted auth permission:

```
python3 /etc/ceph/create-external-cluster-resources.py --cephfs-filesystem-name myfs --rbd-data-pool-name replicapool --cluster-name rookStorage --restricted-auth-permission true
```

Command example of JSON output generated by using the Python script:

```
[{"name": "rook-ceph-mon-endpoints", "kind": "ConfigMap", "data": {"data": "xxx.xxx.xxx.xxx:xxxx", "maxMonId": "0", "mapping": "{}"}}, {"name": "rook-ceph-mon", "kind": "Secret", "data": {"admin-secret": "admin-secret", "fsid": "<fs-id>", "mon-secret": "mon-secret"}}, {"name": "rook-ceph-operator-creds", "kind": "Secret", "data": {"userID": "<user-id>", "userKey": "<user-key>"}}, {"name": "rook-csi-rbd-node", "kind": "Secret", "data": {"userID": "csi-rbd-node", "userKey": "<user-key>"}}, {"name": "ceph-rbd", "kind": "StorageClass", "data": {"pool": "<pool>"}}, {"name": "monitoring-endpoint", "kind": "CephCluster", "data": {"MonitoringEndpoint": "xxx.xxx.xxx.xxx", "MonitoringPort": "xxxx"}}, {"name": "rook-ceph-dashboard-link", "kind": "Secret", "data": {"userID": "ceph-dashboard-link", "userKey": "<user-key>"}}, {"name": "rook-csi-rbd-provisioner", "kind": "Secret", "data": {"userID": "csi-rbd-provisioner", "userKey": "<user-key>"}}, {"name": "rook-csi-cephfs-provisioner", "kind": "Secret", "data": {"adminID": "csi-cephfs-provisioner", "adminKey": "<admin-key>"}}, {"name": "rook-csi-cephfs-node", "kind": "Secret", "data": {"adminID": "csi-cephfs-node", "adminKey": "<admin-key>"}}, {"name": "cephfs", "kind": "StorageClass", "data": {"fsName": "cephfs", "pool": "cephfs_data"}}, {"name": "ceph-rgw", "kind": "StorageClass", "data": {"endpoint": "xxx.xxx.xxx.xxx:xxxx", "poolPrefix": "default"}}, {"name": "rgw-admin-ops-user", "kind": "Secret", "data": {"accessKey": "<access-key>", "secretKey": "<secret-key>"}}]
```

iii. Save the JSON output to a file with .json extension.
Note: For Fusion Data Foundation to work seamlessly, ensure that the parameters (RGW endpoint, CephFS details, RBD pool, and so on) uploaded by using the JSON file remain unchanged on the external IBM Storage Ceph cluster after the storage cluster is created.

iv. Run the command in a multi-tenant deployment where the IBM Storage Ceph cluster is already connected to an IBM Storage Ceph deployment with an earlier version.

```
python3 ceph-external-cluster-details-exporter.py --upgrade
```

c. Click Browse to select and upload the JSON file.
The JSON file content is populated and displayed in the text box.

5. Optional: Select In-transit encryption to enable encryption for entire Ceph traffic including data using Ceph **msgrv2**.

Note: Verify that the external IBM Storage Ceph cluster is configured for in-transit encryption. Before enabling this setting, use the documentation link in the UI to complete the required configuration steps.

a.

i.

1. (Optional) Select the advanced configuration options that apply to your environment:

ii. Review the selections and click **Connect**.

6. Optional: Select the required Advanced configuration option that apply to your environment:
   - Use Ceph RBD as the default StorageClass

     Sets the RBD StorageClass as the cluster default.

   - Use external PostgreSQL *(Technology preview)*

     Uses an external PostgreSQL instance instead of the in-cluster database.

     Note: Additional advanced options may be available in the UI. Select only the options that are required for the deployment policy and environment.
   - Enable Automatic backup

     Allows automatic backup for the Multicloud Object Gateway metadata database.

     - Select the backup frequency: Daily, Weekly, or Monthly.
     - Specify the number of backups to retain.

       Note: This option is disabled if you select Use external PostgreSQL.
7. Review the details, and then click Connect.
8. Validate whether Fusion Data Foundation is successfully configured with an external IBM Storage Ceph cluster in external mode by following the instructions that are mentioned in Verifying Fusion Data Foundation configuration with IBM Storage Ceph.

What to do next

- To add new nodes to increase storage capacity in the cluster, see Scaling out storage capacity.
- To add new disks to a storage node, see Scaling up storage capacity.

- **Verifying Fusion Data Foundation configuration with IBM Storage Ceph**
  Verify that the Fusion Data Foundation is correctly configured with IBM Storage Ceph in external mode.
- **Applying in-transit encryption on IBM Storage Ceph cluster**
  Apply in-transit encryption to enhance data security in your IBM Storage Ceph cluster.

Verifying Fusion Data Foundation configuration with IBM Storage Ceph

Verify that the Fusion Data Foundation is correctly configured with IBM Storage Ceph in external mode.

Procedure

1. Verify the final status of the installed storage cluster.
   a. Go to Storage > Data Foundation > Storage System > ocs-storagecluster.
   b. Verify that the **Status** of the **StorageCluster** is **Ready** and has a green tick mark next to it.
2. Verify the state of the Pods by following the instructions that are mentioned in Verifying the state of the Pods.
3. Verify that the Fusion Data Foundation cluster is healthy by following the instructions that are mentioned in Verifying the Fusion Data Foundation cluster is healthy.
4. Verify that the MCG is healthy by following the instructions that are mentioned in Verifying the Multicloud Object Gateway is healthy.
5. Verify that the Fusion Data Foundation specific storage classes exist by following the instructions that are mentioned in Verifying that the specific storage classes exist.
6. Verify that the IBM Storage Ceph cluster is created and onnected to the external IBM Storage Ceph cluster.
   Command example:

   ```
   oc get cephcluster -n openshift-storage
   ```

   Example output:

   ```
   NAME                                       DATADIRHOSTPATH   MONCOUNT   AGE   PHASE       MESSAGE                          HEAL
   ocs-external-storagecluster-cephcluster                                 30m   Connected   Cluster connected successfully   HEAL
   ```

7. Verify that the Fusion Data Foundation cluster is ready and the **External** option is set to *true*.
   Command example:

   ```
   oc get cephcluster -n openshift-storage
   ```

   Example output:

   ```
   NAME                          AGE   PHASE   EXTERNAL   CREATED AT             VERSION
   ocs-external-storagecluster   30m   Ready   true       2021-11-17T09:09:52Z   4.19.0
   ```

8. Verify that the Ceph Object Store CRD is created in the IBM Storage Ceph cluster.
   Command example:

   ```
   oc get cephobjectstore -n openshift-storage
   ```

   Example output:

   ```
   NAME               PHASE     ENDPOINT               SECUREENDPOINT      AGE
   object-store1      Ready     <http://IP/FQDN:port>                      15m
   ```

- **Verifying the state of the Pods**
  Validate the state of Pods to ensure that all components are running correctly and the Fusion Data Foundation cluster is healthy.
- **Verifying the Fusion Data Foundation cluster is healthy**
  Check the health of the Fusion Data Foundation cluster to ensure Fusion Data Foundation is operating as expected.
- **Verifying the Multicloud Object Gateway is healthy**
  Check the health status of the Multicloud Object Gateway (MCG) to confirm its availability in the Fusion Data Foundation cluster.
- **Verifying that the specific storage classes exist**
  Confirm that necessary storage classes are available in the Fusion Data Foundation cluster to support workloads and maintain consistent storage operations.

Verifying the state of the Pods

Validate the state of Pods to ensure that all components are running correctly and the Fusion Data Foundation cluster is healthy.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, select openshift-storage from the Project drop-down list.
   Note: If the Show default projects option is disabled, use the toggle button to list all the default projects.
2. Go to Workloads > Pods.
   Set filter for running and completed pods to verify that the following pods are in **`Running`** and **`Completed`** state:

Table 1. Pods corresponding to Fusion Data Foundation

Component	Pods
Fusion Data Foundation Operator	• **`ocs-operator-`*** (1 pod on any storage node) • **`ocs-metrics-exporter-`*** (1 pod on any storage node) • **`odf-operator-controller-manager-`*** (1 pod on any storage node) • **`odf-console-`*** (1 pod on any storage node) • **`csi-addons-controller-manager-`*** (1 pod on any storage node)
Rook-ceph Operator	**`rook-ceph-operator-`*** (1 pod on any storage node)
Multicloud Object Gateway	• **`noobaa-operator-`*** (1 pod on any storage node) • **`noobaa-core-`*** (1 pod on any storage node) • **`noobaa-db-pg-cluster-1`*** and **`noobaa-db-pg-cluster-2`*** (2 instances of MCG DB pod on any storage node) • **`noobaa-endpoint-`*** (1 pod on any storage node) • **`cnpg-controller-manager-`*** (1 pod on any storage node)

CSI	• **`cephfs`**
	  ◦ **`openshift-storage.cephfs.csi.ceph.com-ctrlplugin-`*** (2 pods distributed across storage nodes)
	  ◦ **`openshift-storage.cephfs.csi.ceph.com-nodeplugin-`*** (1 pod on each storage node)
	Note: If an MDS is not deployed in the external cluster, the **`openshift-storage.cephfs.csi.ceph.com-ctrlplugin`** and **`openshift-storage.cephfs.csi.ceph.com-nodeplugin-`*** pods are not created.
	• **`nfs`**
	  ◦ **`openshift-storage.nfs.csi.ceph.com-ctrlplugin-`*** (2 pods distributed across storage nodes)
	  ◦ **`openshift-storage.nfs.csi.ceph.com-nodeplugin-`*** (1 pod on each storage node)
	• **`rbd`**
	  ◦ **`openshift-storage.rbd.csi.ceph.com-ctrlplugin-`*** (2 pods distributed across storage nodes)
	  ◦ **`openshift-storage.rbd.csi.ceph.com-nodeplugin-`*** (1 pod on each storage node)

Verifying the Fusion Data Foundation cluster is healthy

Check the health of the Fusion Data Foundation cluster to ensure Fusion Data Foundation is operating as expected.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. In the Status card of the Block and File tab, verify that the Storage Cluster has a green tick.
4. In the Details card, verify that the cluster information is displayed.
   For more information on the health of the Fusion Data Foundation cluster using the Block and File dashboard, see Monitoring Fusion Data Foundation.

Verifying the Multicloud Object Gateway is healthy

Check the health status of the Multicloud Object Gateway (MCG) to confirm its availability in the Fusion Data Foundation cluster.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. In the Status card of the Object tab, verify that both Object Service and Data Resiliency have a green tick.

4. In the Details card, verify that the MCG information is displayed.
   For more information on the health of the Fusion Data Foundation cluster using the object service dashboard, see Monitoring Fusion Data Foundation.

   Note:
   The RADOS Object Gateway appears only when its endpoint details are included during the Fusion Data Foundation configuration in external mode.

Verifying that the specific storage classes exist

Confirm that necessary storage classes are available in the Fusion Data Foundation cluster to support workloads and maintain consistent storage operations.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Storage Classes.
2. Verify that the following storage classes are created with the Fusion Data Foundation cluster creation:
   - `ocs-storagecluster-ceph-rbd`
   - `ocs-storagecluster-ceph-rgw`
   - `ocs-storagecluster-cephfs`
   - `openshift-storage.noobaa.io`

   Note:
   - If an MDS is not deployed in the external cluster, the `ocs-external-storagecluster-cephfs` storage class is not created.
   - If an RGW is not deployed in the external cluster, the `ocs-external-storagecluster-ceph-rgw` storage class is not created.

   For more information on MDS and RGW, see IBM Storage Ceph documentation.

Applying in-transit encryption on IBM Storage Ceph cluster

Apply in-transit encryption to enhance data security in your IBM Storage Ceph cluster.

Procedure

1. Apply in-transit encryption settings.
   Command example:

   ```
   ceph config set global ms_client_mode secure
   ceph config set global ms_cluster_mode secure
   ceph config set global ms_service_mode secure
   ceph config set global rbd_default_map_options ms_mode=secure
   ```

2. Check the settings.
   Command example:

   ```
   ceph config dump | grep ms_
   ```

   Example output:

   ```
   global                    basic     ms_client_mode                secure                *
   global                    basic     ms_cluster_mode               secure                *
   global                    basic     ms_service_mode               secure                *
   global                    advanced  rbd_default_map_options        ms_mode=secure        *
   ```

3. Restart all Ceph daemons.
   Command example:

   ```
   ceph orch ps
   ```

   Example output:

   ```
   NAME                      HOST   PORTS             STATUS         REFRESHED  AGE  MEM USE  MEM LIM  VERSION           IMAGE ID
   alertmanager.osd-0        osd-0  *:9093,9094       running (7h)     5m ago   7h    24.6M        -  0.24.0            3d2ad4f3
   ceph-exporter.osd-0       osd-0                    running (7h)     5m ago   7h    17.7M        -  18.2.0-192.el9cp  6e4e34f0
   ceph-exporter.osd-1       osd-1                    running (7h)     5m ago   7h    17.8M        -  18.2.0-192.el9cp  6e4e34f0
   ceph-exporter.osd-2       osd-2                    running (7h)     5m ago   7h    17.9M        -  18.2.0-192.el9cp  6e4e34f0
   ceph-exporter.osd-3       osd-3                    running (7h)     5m ago   7h    17.7M        -  18.2.0-192.el9cp  6e4e34f0
   crash.osd-0               osd-0                    running (7h)     5m ago   7h    6895k        -  18.2.0-192.el9cp  6e4e34f0
   crash.osd-1               osd-1                    running (7h)     5m ago   7h    6895k        -  18.2.0-192.el9cp  6e4e34f0
   crash.osd-2               osd-2                    running (7h)     5m ago   7h    6903k        -  18.2.0-192.el9cp  6e4e34f0
   crash.osd-3               osd-3                    running (7h)     5m ago   7h    6899k        -  18.2.0-192.el9cp  6e4e34f0
   grafana.osd-0             osd-0  *:3000            running (7h)     5m ago   7h    65.5M        -  <unknown>         f142b583
   mds.fsvol001.osd-0.lpciqk osd-0                    running (7h)     5m ago   7h    24.8M        -  18.2.0-192.el9cp  6e4e34f0
   mds.fsvol001.osd-2.wocnxz osd-2                    running (7h)     5m ago   7h    32.1M        -  18.2.0-192.el9cp  6e4e34f0
   mgr.osd-0.dtkyni          osd-0  *:9283,8765,8443  running (7h)     5m ago   7h     535M        -  18.2.0-192.el9cp  6e4e34f0
   mgr.osd-2.kqcxwu          osd-2  *:8443,9283,8765  running (7h)     5m ago   7h     440M        -  18.2.0-192.el9cp  6e4e34f0
   mon.osd-1                 osd-1                    running (7h)     5m ago   7h     350M    2048M  18.2.0-192.el9cp  6e4e34f0
   mon.osd-2                 osd-2                    running (7h)     5m ago   7h     363M    2048M  18.2.0-192.el9cp  6e4e34f0
   mon.osd-3                 osd-3                    running (7h)     5m ago   7h     361M    2048M  18.2.0-192.el9cp  6e4e34f0
   node-exporter.osd-0       osd-0  *:9100            running (7h)     5m ago   7h    25.1M        -  1.4.0             508050f8
   node-exporter.osd-1       osd-1  *:9100            running (7h)     5m ago   7h    21.4M        -  1.4.0             508050f8
   node-exporter.osd-2       osd-2  *:9100            running (7h)     5m ago   7h    25.4M        -  1.4.0             508050f8
   node-exporter.osd-3       osd-3  *:9100            running (7h)     5m ago   7h    21.8M        -  1.4.0             508050f8
   osd.0                     osd-2                    running (7h)     5m ago   7h     525M    4096M  18.2.0-192.el9cp  6e4e34f0
   osd.1                     osd-0                    running (7h)     5m ago   7h     652M    4096M  18.2.0-192.el9cp  6e4e34f0
   osd.2                     osd-3                    running (7h)     5m ago   7h     801M    1435M  18.2.0-192.el9cp  6e4e34f0
   ```

```
osd.3                       osd-1                    running (7h)     5m ago   7h     538M     923M  18.2.0-192.el9cp  6e4e34f0
osd.4                       osd-0                    running (7h)     5m ago   7h     532M    4096M  18.2.0-192.el9cp  6e4e34f0
osd.5                       osd-2                    running (7h)     5m ago   7h     761M    4096M  18.2.0-192.el9cp  6e4e34f0
osd.6                       osd-3                    running (7h)     5m ago   7h     415M    1435M  18.2.0-192.el9cp  6e4e34f0
osd.7                       osd-1                    running (7h)     5m ago   7h     427M     923M  18.2.0-192.el9cp  6e4e34f0
prometheus.osd-0            osd-0  *:9095            running (7h)     5m ago   7h    84.0M       -  2.39.1            716dd9df
rgw.rgw.ssl.osd-1.smzpfj    osd-1  *:80              running (7h)     5m ago   7h     110M       -  18.2.0-192.el9cp  6e4e34f0
```

Wait for all the daemons to restart.

Configuring with IBM FlashSystem

Configure Fusion Data Foundation in external mode with an external IBM FlashSystem cluster.

Before you begin

Fusion Data Foundation service is installed with device type set to External, as instructed in Installing Fusion Data Foundation.

Procedure

1. Log in to the Red Hat® OpenShift® Container Platform web console as an administrator, and navigate to Storage > External systems.
2. Click Connect external systems.
3. Select IBM FlashSystem and click Next.
4. On the Create storage class tab, do as follows:
   a. Enter a name for the storage class.
      Remember: The storage class name must not exceed 40 characters and must not match any of the following predefined storage classes:
      - `ocs-storagecluster-cephfs`
      - `ocs-storagecluster-ceph-rbd`
      - `ocs-storagecluster-ceph-rgw`
      - `openshift-storage.noobaa.io`
      - `ocs-storagecluster-ceph-rbd-virtualization`

      When creating block storage persistent volumes, select the `<storage_class_name>` storage class for optimal performance. This storage class provides a direct I/O path to the FlashSystem.

   b. Enter the following details of the IBM FlashSystem connection.
      - IP address
      - User name
      - Password
      - Pool name

   c. Select `thick` or `thin` for the Volume mode.
   d. Click Next.
5. On the Capacity and nodes tab, do as follows:
   a. Select a value from the Requested capacity list.
      The available options are 0.5 TiB, 2 TiB, and 4 TiB. The requested capacity is dynamically allocated on the infrastructure storage class.
   b. Select at least three nodes in three different zones.
      Start with at least 14 CPUs and 34 GiB of RAM per node. If the selected nodes do not meet the aggregated requirement of 30 CPUs and 72 GiB of RAM for Fusion Data Foundation, a minimal cluster is deployed instead. For minimum starting node requirements, see Resource requirements.
   c. Click Next.
6. Optional: On the Security and network tab, do as follows:
   a. To enable encryption, select Enable data encryption for block and file storage.
      i. Choose any one or both encryption level:
         - Cluster-wide encryption to encrypt the entire cluster (block and file).
         - StorageClass encryption to create encrypted persistent volume (block only) by using encryption-enabled storage class.

      ii. Select the Connect to an external key management service checkbox, which is optional for cluster-wide encryption.
         1. Key Management Service Provider is set to Vault by default.
         2. Enter the details for the Vault Service Name, Host Address of Vault server ('https://<hostname or ip>'), Port number, and Token fields.

      iii. Expand Advanced Settings to enter the following details based on your vault configuration:
         1. Enter the key value secret path that is dedicated and unique to Fusion Data Foundation in the Backend Path field.
         2. Optional: Enter the details for the TLS Server Name and Vault Enterprise Namespace fields.
         3. Upload the respective PEM-encoded certificate file to fill the CA Certificate, Client Certificate, and Client Private Key fields.

      Note: In case you need to enable key rotation for Vault KMS, run the following command in the Red Hat OpenShift Container Platform web console after the storage cluster is created:

      ```
      oc patch storagecluster ocs-storagecluster -n openshift-storage --type=json -p '[{"op": "add",
      "path":"/spec/encryption/keyRotation/enable", "value": true}]'
      ```

      iv. Click Save.
      v. Select Default (SDN) for a single network and Custom (Multus) for multiple network interfaces.
         1. Select an interface from the Public Network Interface list.
         2. Select an interface from the Cluster Network Interface list.
            Note: If you are using only one additional network interface, select the single NetworkAttachementDefinition, that is, ocs-public-cluster for the Public Network Interface field, and leave the Cluster Network Interface field empty.

      vi. Click Next.
   b. If you select In-transit encryption, add the details in the Network field and click Next.
7. On the Review and create tab, review the configuration details.

To modify any configuration settings, click Back to go back to the previous configuration page.

8. Click Create StorageSystem.
9. Validate whether Fusion Data Foundation is successfully configured with an external IBM FlashSystem cluster in external mode by following the instructions that are mentioned in Verifying Fusion Data Foundation configuration with IBM FlashSystem.

What to do next

- To add new nodes to increase storage capacity in the cluster, see Scaling out storage capacity.
- To add new disks to a storage node, see Scaling up storage capacity.

- **Verifying Fusion Data Foundation configuration with IBM FlashSystem**
  Verify that the Fusion Data Foundation is correctly configured with IBM FlashSystem in external mode.

Verifying Fusion Data Foundation configuration with IBM FlashSystem

Verify that the Fusion Data Foundation is correctly configured with IBM FlashSystem in external mode.

Procedure

1. Verify the state of the Pods by following the instructions that are mentioned in Verifying the state of the Pods.
2. Verify that the Fusion Data Foundation cluster is healthy by following the instructions that are mentioned in Verifying the Fusion Data Foundation cluster is healthy.
3. Verify that the MCG is healthy by following the instructions that are mentioned in Verifying the Multicloud Object Gateway is healthy.
4. Verify that the IBM FlashSystem is connected and the storage cluster is ready.
   Command example:

   ```
   oc get flashsystemclusters.odf.ibm.com
   ```

   Example output:

   ```
   NAME                              AGE   PHASE   CREATED AT
   ibm-flashsystem-storage-fab3p-60   46h   Ready   2025-06-30T12:17:06Z
   ```

5. Verify the storage system of the IBM FlashSystem cluster.
   Command example:

   ```
   oc get storagesystems.odf.openshift.io
   ```

   Example output:

   ```
   NAME                                  STORAGE-SYSTEM-KIND                       STORAGE-SYSTEM-NAME
   ibm-flashsystemcluster-storagesystem  flashsystemcluster.odf.ibm.com/v1alpha1   ibm-flashsystemcluster
   ocs-storagecluster                    storagecluster.ocs.openshift.io/v1        ocs-storagecluster
   ```

6. Verify the subscription of the IBM operator.
   Command example:

   ```
   oc get subscriptions.operators.coreos.com
   ```

   Example output:

   ```
   NAME                                                                                PACKAGE                    SOURCE
   cephcsi-operator-stable-4.19-redhat-operators-openshift-marketplace                 cephcsi-operator           redhat-operators
   ibm-block-csi-operator-stable-certified-operators-openshift-marketplace             ibm-block-csi-operator     certified-operators
   ibm-storage-odf-operator                                                            ibm-storage-odf-operator   ibm-storage-odf-catalo
   mcg-operator-stable-4.19-redhat-operators-openshift-marketplace                     mcg-operator               redhat-operators
   ocs-client-operator-stable-4.19-redhat-operators-openshift-marketplace              ocs-client-operator        redhat-operators
   ocs-operator-stable-4.19-redhat-operators-openshift-marketplace                     ocs-operator               redhat-operators
   odf-csi-addons-operator-stable-4.19-redhat-operators-openshift-marketplace          odf-csi-addons-operator    redhat-operators
   odf-dependencies                                                                    odf-dependencies           redhat-operators
   odf-operator                                                                        odf-operator               redhat-operators
   odf-prometheus-operator-stable-4.19-redhat-operators-openshift-marketplace          odf-prometheus-operator    redhat-operators
   recipe-stable-4.19-redhat-operators-openshift-marketplace                           recipe                     redhat-operators
   rook-ceph-operator-stable-4.19-redhat-operators-openshift-marketplace               rook-ceph-operator         redhat-operators
   ```

7. Verify that the CSVs are in the succeeded state.
   Command example:

   ```
   oc get csv
   ```

   Example output:

   ```
   NAME                                  DISPLAY                                VERSION        REPLACES
   cephcsi-operator.v4.19-rhodf          CephCSI operator                       4.19-rhodf     cephcsi-operator.v4.19-rhodf
   ibm-block-csi-operator.v1.12.4        IBM block storage CSI driver operator  1.12.4
   ibm-storage-odf-operator.v1.7.1       IBM Storage ODF operator               1.7.1          ibm-storage-odf-operator.v1.7.0
   mcg-operator.v4.19-rhodf              NooBaa Operator                        4.19-rhodf     mcg-operator.v4.19-rhodf
   ocs-client-operator.v4.19-rhodf       OpenShift Data Foundation Client       4.19-rhodf     ocs-client-operator.v4.19-rhodf
   ocs-operator.v4.19-rhodf              OpenShift Container Storage            4.19-rhodf     ocs-operator.v4.19-rhodf
   odf-csi-addons-operator.v4.19-rhodf   CSI Addons                             4.19-rhodf     odf-csi-addons-operator.v4.19-rho
   odf-dependencies.v4.19-rhodf          Data Foundation Dependencies           4.19-rhodf     odf-dependencies.v4.19-rhodf
   odf-operator.v4.19-rhodf              OpenShift Data Foundation              4.19-rhodf     odf-operator.v4.19-rhodf
   odf-prometheus-operator.v4.19-rhodf   Prometheus Operator                    4.19-rhodf     odf-prometheus-operator.v4.19-rho
   recipe.v4.19-rhodf                    Recipe                                 4.19-rhodf     recipe.v4.19-rhodf
   rook-ceph-operator.v4.19-rhodf        Rook-Ceph
   ```

8. Verify the IBM operator and CSI Pods.

Command example:

```
oc get pods
```

Example output:

```
NAME                                                              READY   STATUS      RESTARTS   AGE
ceph-csi-controller-manager-65f4bc4cd5-9qbp4                      2/2     Running     0          46h
csi-addons-controller-manager-757d9c6748-94nrt                    2/2     Running     0          46h
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-b642k            3/3     Running     0          46h
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-l75pt            3/3     Running     0          46h
openshift-storage.cephfs.csi.ceph.com-nodeplugin-847f7c54d8       7/7     Running     0          46h
openshift-storage.cephfs.csi.ceph.com-nodeplugin-847f7c54b9       7/7     Running     0          46h
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-vh485            3/3     Running     0          46h
openshift-storage.rbd.csi.ceph.com-ctrlplugin-9hss7               4/4     Running     0          46h
openshift-storage.rbd.csi.ceph.com-ctrlplugin-9q9qd               4/4     Running     0          46h
openshift-storage.rbd.csi.ceph.com-ctrlplugin-9vkqh               4/4     Running     0          46h
openshift-storage.rbd.csi.ceph.com-nodeplugin-7f4b8d946f-ngz65    7/7     Running     0          46h
openshift-storage.rbd.csi.ceph.com-nodeplugin-7f4b8d946f-x818t    7/7     Running     0          46h
ibm-block-csi-controller-0                                        8/8     Running     0          41h
ibm-block-csi-node-6vmpq                                          3/3     Running     0          46h
ibm-block-csi-node-fw7xc                                          3/3     Running     0          41h
ibm-block-csi-node-k57sv                                          3/3     Running     0          46h
ibm-block-csi-operator-867c54df5d-2t968                           1/1     Running     0          46h
ibm-flashsystem-storage-deployment-5958ffd56b-tj6zw               1/1     Running     0          46h
ibm-odf-console-6d64dcc46c-bkknx                                  1/1     Running     0          41h
ibm-storage-odf-operator-787c67c7bf-nzp9w                         2/2     Running     0          41h
noobaa-core-0                                                     2/2     Running     0          46h
noobaa-db-pg-0                                                    1/1     Running     0          46h
noobaa-endpoint-5d94d888fc-bj28x                                  1/1     Running     0          46h
noobaa-operator-67b569fc76-rk99d                                  1/1     Running     0          46h
ocs-metrics-exporter-6f7f79fcf9-xm6pq                             2/3     Running     0          46h
ocs-operator-676c964c58-nsb7g                                     1/1     Running     0          46h
odf-console-75dc785696-n97sk                                      1/1     Running     0          46h
odf-operator-controller-manager-785ddcd999-tf245                  2/2     Running     0          46h
rook-ceph-crashcollector-worker-node1-b95b5cd54-7qbhb             1/1     Running     0          46h
rook-ceph-crashcollector-worker-node2-757d746d5f-jsfrx            1/1     Running     0          46h
rook-ceph-crashcollector-worker-node3-6d4f8f7848-xvvw2            1/1     Running     0          46h
rook-ceph-exporter-worker-node1-5dcf74b647-t5rg7                  1/1     Running     0          46h
rook-ceph-exporter-worker-node2-66c694f5b6-n96g5                  1/1     Running     0          46h
rook-ceph-exporter-worker-node3-58fd7ffcfc-vpzfr                  1/1     Running     0          46h
rook-ceph-mds-ocs-storagecluster-cephfilesystem-a-557b4cc9gsx65   2/2     Running     0          46h
rook-ceph-mds-ocs-storagecluster-cephfilesystem-b-5d8748dfmrhfn   2/2     Running     0          46h
rook-ceph-mgr-a-f56c6866b-tzl2s                                   3/3     Running     0          46h
rook-ceph-mgr-b-796d5f69bf-f54qr                                  3/3     Running     0          46h
rook-ceph-mon-a-6b5dc9dd6c-f4dvn                                  2/2     Running     0          46h
rook-ceph-mon-b-8d458fbdf-wjgzm                                   2/2     Running     0          46h
rook-ceph-mon-c-5b78749887-mcpjl                                  2/2     Running     0          46h
rook-ceph-operator-578dfdd8dd-mn9vn                               1/1     Running     0          46h
rook-ceph-osd-0-754499c4bd-2nrs2                                  2/2     Running     0          46h
rook-ceph-osd-1-5566ff6d99-dc614                                  2/2     Running     0          46h
rook-ceph-osd-2-8567c9f9f7-wr66b                                  2/2     Running     0          46h
rook-ceph-osd-prepare-667012a10f7f0800994523ce6731b5d1-n7fgb       0/1     Completed   0          46h
rook-ceph-osd-prepare-ab6be2d13c76aad7123b793996562bf0-wpqsv      0/1     Completed   0          46h
rook-ceph-osd-prepare-baf6cafc480b4e046c337c49601633b8-zllnx      0/1     Completed   0          46h
rook-ceph-rgw-ocs-storagecluster-cephobjectstore-a-5f4dfd9wlbvn   2/2     Running     0          46h
ux-backend-server-db4c5b7bd-hn9jx                                 2/2     Running     0          46h
```

- **Verifying the state of the Pods**
  Validate the state of Pods to ensure that all components are running correctly and the Fusion Data Foundation cluster is healthy.
- **Verifying the Fusion Data Foundation cluster is healthy**
  Check the health of the Fusion Data Foundation cluster to ensure Fusion Data Foundation is operating as expected.
- **Verifying the Multicloud Object Gateway is healthy**
  Check the health status of the Multicloud Object Gateway (MCG) to confirm its availability in the Fusion Data Foundation cluster.

Verifying the state of the Pods

Validate the state of Pods to ensure that all components are running correctly and the Fusion Data Foundation cluster is healthy.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, select openshift-storage from the Project drop-down list.
   Note: If the Show default projects option is disabled, use the toggle button to list all the default projects.
2. Go to Workloads > Pods.
   Set filter for running and completed pods to verify that the following pods are in **Running** and **Completed** state:

Table 1. Pods corresponding to Fusion Data Foundation

Component	Pods
Fusion Data Foundation Operator	• **ocs-operator-*** (1 pod on any storage node) • **ocs-metrics-exporter-*** (1 pod on any storage node) • **odf-operator-controller-manager-*** (1 pod on any storage node) • **odf-console-*** (1 pod on any storage node) • **csi-addons-controller-manager-*** (1 pod on any storage node)
ibm-storage-odf-operator	• **ibm-storage-odf-operator-*** (2 pods on any worker nodes) • **ibm-odf-console-***

Component	Pods
ibm-flashsystem-storage	`ibm-flashsystem-storage-*` (1 pod on any worker node)
rook-ceph Operator	`rook-ceph-operator-*` (1 pod on any worker node)
Multicloud Object Gateway	• `noobaa-operator-*` (1 pod on any storage node) • `noobaa-core-*` (1 pod on any storage node) • `noobaa-db-pg-cluster-1*` and `noobaa-db-pg-cluster-2*` (2 instances of MCG DB pod on any storage node) • `noobaa-endpoint-*` (1 pod on any storage node) • `cnpg-controller-manager-*` (1 pod on any storage node)
CSI	`ibm-block-csi-*` (1 pod on any worker node)

Verifying the Fusion Data Foundation cluster is healthy

Check the health of the Fusion Data Foundation cluster to ensure Fusion Data Foundation is operating as expected.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. In the Status card of the Block and File tab, verify that the Storage Cluster has a green tick.
4. In the Details card, verify that the cluster information is displayed.
   For more information on the health of the Fusion Data Foundation cluster using the Block and File dashboard, see Monitoring Fusion Data Foundation.

Verifying the Multicloud Object Gateway is healthy

Check the health status of the Multicloud Object Gateway (MCG) to confirm its availability in the Fusion Data Foundation cluster.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. In the Status card of the Object tab, verify that both Object Service and Data Resiliency have a green tick.
4. In the Details card, verify that the MCG information is displayed.
   For more information on the health of the Fusion Data Foundation cluster using the object service dashboard, see Monitoring Fusion Data Foundation.

   Note:
   The RADOS Object Gateway appears only when its endpoint details are included during the Fusion Data Foundation configuration in external mode.

Configuring with IBM Storage Scale

Configure IBM Storage Scale (CNSA) to integrate fast, scalable, and reliable file storage with Fusion Data Foundation, ensuring high performance and reliability for enterprise workloads.

Before you begin

Ensure that the following prerequisites are met before configuring IBM Storage Scale (CNSA) in the Red Hat® OpenShift® Container Platform web console:

- Fusion Data Foundation is installed with the device type set to **External**.

  For instructions, see Installing Fusion Data Foundation.

- The **MachineConfig Operator (MCO)** required for IBM Storage Scale is applied.

  Command example:

  ```
  kubectl apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v6.0.0.x/generated/scale/mco/mco.yaml
  ```

  If you are running OpenShift on s390x, apply the following MachineConfig instead:

  ```
  kubectl apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v6.0.0.x/generated/scale/mco/mco_s390x.yaml
  ```

  Wait 15–20 minutes for the MachineConfig changes to take effect.

- Verify that all machine config pools are updated:

  Command example:

  ```
  oc get mcp
  ```

  Ensure that the **UPDATED** column shows **True** for all entries.

- An **IBM entitlement key** is available.

For instructions on obtaining the entitlement key, see Obtaining your entitlement key in *IBM Cloud Paks* documentation.

- An **ibm-entitlement-key** secret is created in the **ibm-spectrum-scale** namespace using the entitlement key.

  Command example:

```
NS=ibm-spectrum-scale
# 1) Extract the entitlement key (token string)
ENT_KEY=YOUR_KEY
echo "$ENT_KEY"
# 2) Build a valid docker config JSON using that token
AUTH_B64=$(printf 'cp:%s' "$ENT_KEY" | base64 | tr -d '\n')
cat > /tmp/ibm-dockerconfig.json <<EOF
{
  "auths": {
    "cp.icr.io": {
      "username": "cp",
      "password": "$ENT_KEY",
      "auth": "$AUTH_B64"
    }
  }
}
EOF
# 3) Create/update ibm-entitlement-key as dockerconfigjson
oc create secret generic ibm-entitlement-key \
  -n "$NS" \
  --from-file=.dockerconfigjson=/tmp/ibm-dockerconfig.json \
  --type=kubernetes.io/dockerconfigjson \
  --dry-run=client -o yaml | oc apply -f -
```

  Important: Replace `YOUR_KEY` with your **IBM entitlement key**.
- Host aliases for all virtual machines are added to the **ibm-spectrum-scale** cluster custom resource under `spec.daemon`.

  Command example:

```
oc -n ibm-spectrum-scale edit cluster.scale.spectrum.ibm.com/ibm-spectrum-scale
```

- The following resources are added to the **ibm-spectrum-scale** cluster custom resource:
  - CPU: 2
  - Memory: 6Gi

  YAML example of **ibm-spectrum-scale** cluster custom resource:

```
    bypassDiscovery: true
  resources:
    requests:
      cpu: '2'
      memory: 6Gi
  roles: []
```

Procedure

1. Log in to the Red Hat OpenShift Container Platform web console as an administrator, and navigate to Storage > External systems.
2. Click Connect external systems.
3. Select IBM Scale and click Next.
   The Connect IBM Scale (CNSA) page appears.
4. In the Name field, enter a name for the cluster.
   Remember: The name must meet the following requirements:
   - It is 3–63 characters long.
   - It starts and ends with a lowercase letter or number.
   - It includes only lowercase letters, numbers, nonconsecutive periods, or hyphens.
   - It is unique.
5. Select local cluster nodes.
   You can choose one of the following options:

   All nodes
   : If you choose this option, all non control plane nodes in this infrastructure are automatically selected to handle requests to IBM Storage Scale.

   Select nodes
   : If you choose this option, you must select a minimum of three nodes to handle requests to IBM Storage Scale.

   Note: The All nodes option is selected by default.
   If you choose the Select nodes option, select at least three nodes from the node list.
   Tip: You can search for nodes by name.
6. Add the following connection details:
   a. In the Management endpoints field, enter the GUI hostname or IP address of the IBM Storage Scale management endpoint.
   b. In the Port field, enter the port number used by the management endpoint.
   c. Optional: To configure additional endpoints for high availability:
      - Enter the hostname or IP address in Optional endpoint 1 and specify the corresponding port number in the Optional port 1 field.
      - Enter the hostname or IP address in Optional endpoint 2 and specify the corresponding port number in the Optional port 2 field.

   d. In the User name field, enter a user name that is 1–63 characters long.
   e. In the Password field, enter a password.
   f. Optional: In the CA certificate field, browse and upload a certificate to secure the configuration.
7. Add the following file system configuration details:
   a. In the File system name field, enter a name for the remote file system to access on this cluster.
      Remember: The name must meet the following requirements:
      - It is 3–63 characters long.

- It starts and ends with a lowercase letter or number.
- It includes only lowercase letters, numbers, nonconsecutive periods, or hyphens.
- It is unique.

b. Optional: Select the Enable data encryption checkbox to ensure that all filesystem data is securely stored and protected.
If you select the Enable data encryption checkbox, enter the following mandatory details:
- Username
- Port
- Client
- Remote RKM
- Encryption CA certificate
- Server information
- Tenant ID

8. Click Connect.
Note: The Scale cluster might take a few minutes to initialize.
The Scale cluster appears in the External systems page.

9. Verify the Scale cluster configuration:
a. Click the Scale cluster name to open the Scale Dashboard.
b. In the Scale Dashboard, check the status of the Scale cluster.
Note: If you entered incorrect data during first time IBM Storage Scale configuration and need to delete the invalid Scale cluster, see Unable to delete incorrect IBM Scale (CNSA) cluster after first-time configuration.

Results

IBM Storage Scale (CNSA) cluster is integrated with Fusion Data Foundation.

What to do next

After configuring the IBM Storage Scale (CNSA) cluster, you can add more file systems for use in the cluster. For instructions, see Adding file systems to an IBM Storage Scale (CNSA) cluster.

To remove a remote IBM Storage Scale (CNSA) cluster, see Removing a remote IBM Storage Scale (CNSA) cluster.

- **Adding file systems to an IBM Storage Scale (CNSA) cluster**
Provides steps to add multiple remote file systems to an existing IBM Storage Scale (CNSA) cluster using the Red Hat OpenShift Container Platform web console.
- **Removing a filesystem from an IBM Storage Scale (CNSA) cluster**
Remove a remote filesystem from an IBM Storage Scale (CNSA) cluster using the Red Hat OpenShift Container Platform web console.
- **Removing a remote IBM Storage Scale (CNSA) cluster**
Remove a remote IBM Storage Scale (CNSA) cluster that is configured as an external system.

Adding file systems to an IBM Storage Scale (CNSA) cluster

Provides steps to add multiple remote file systems to an existing IBM Storage Scale (CNSA) cluster using the Red Hat® OpenShift® Container Platform web console.

Before you begin

Ensure that the following prerequisites are met:

- An IBM Storage Scale (CNSA) cluster is already configured as an external system. For instructions, see Configuring with IBM Storage Scale.
- You have administrative access to the Red Hat OpenShift Container Platform web console.
- The file systems to be added already exist on the IBM Storage Scale cluster.

Procedure

1. Log in to the Red Hat OpenShift Container Platform web console with administrative privileges.
2. From the navigation menu, go to Storage > External systems.
3. On the External systems page, locate the configured IBM Scale (CNSA) cluster.
4. Click the More actions (⋮) icon at the end of the IBM Scale (CNSA) cluster row.
5. Select Add Remote FileSystem.
6. In the Add Remote FileSystem dialog, enter the name of the IBM Storage Scale file system that you want to add.
7. Click Add.
Note: To exit without adding the file system, click Cancel.
8. Verify that the newly added file system appears under the selected IBM Storage Scale (CNSA) cluster.
9. Repeat steps 4 through 8 to add additional file systems to the same IBM Storage Scale (CNSA) cluster.
10. Create a StorageClass for each added file system, if required.
After adding the file system, manually create a StorageClass to enable dynamic provisioning.

Important: In IBM Fusion 2.12 and earlier releases, IBM Fusion automatically created a StorageClass when a file system was added. Starting with IBM Fusion 2.13 and later, and with the integration of the IBM Storage Scale remote mount capability in Fusion Data Foundation 4.21 or later, StorageClasses are no longer created automatically. You must create them manually after adding the file system, if required.
Use the following example as a reference to create a StorageClass for each added file system. Update the parameter values to match your environment and file system configuration.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-sc
```

```
provisioner: spectrumscale.csi.ibm.com
parameters:
  clusterId: "1234567890123456"
  volBackendFs: "fs1-1234567890123456"
  version: "2"
  shared: "false"
allowVolumeExpansion: true
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

For more information about the StorageClass configurations supported by IBM Storage Scale, see Configuring a storage class for use with the CSI driver.

Removing a filesystem from an IBM Storage Scale (CNSA) cluster

Remove a remote filesystem from an IBM Storage Scale (CNSA) cluster using the Red Hat® OpenShift® Container Platform web console.

Before you begin

Ensure that the following prerequisites are met:

- An IBM Storage Scale (CNSA) cluster is already configured as an external system. For instructions, see Configuring with IBM Storage Scale.
- At least one remote filesystem is already added to the cluster. For instructions, see Adding file systems to an IBM Storage Scale (CNSA) cluster.
- You have administrative access to the Red Hat OpenShift Container Platform web console.

Procedure

1. List all filesystems connected to the IBM Storage Scale cluster.
   Command example:

   ```
   oc get filesystem -n ibm-spectrum-scale
   ```

2. Delete the filesystem.
   Command example:

   ```
   oc delete filesystem <filesystem_name> -n ibm-spectrum-scale
   ```

3. Verify that the filesystem no longer appears in the IBM Storage Scale cluster.
   Note: Deleting the filesystem custom resource does not remove internal GPFS `remotecluster` definitions. For internal GFPS resources removal, see Filesystem deletion does not remove internal IBM Storage Scale resources.

Results

You removed the remote filesystem from the IBM Storage Scale (CNSA) cluster.

Removing a remote IBM Storage Scale (CNSA) cluster

Remove a remote IBM Storage Scale (CNSA) cluster that is configured as an external system.

Before you begin

Ensure that the following prerequisites are met:

- Stop all applications that are accessing storage through the IBM Storage Scale CSI driver.
- Delete all Persistent Volume Claims (PVCs) and Persistent Volumes (PVs) that were provisioned by the IBM Storage Scale CSI driver.

Procedure

1. Delete the remote file systems from the IBM Storage Scale cluster.
   For instructions, see Remote file systems in the *IBM Storage Scale Container Native* documentation.

2. Delete the remote cluster from the IBM Storage Scale cluster.
   For instructions, see Remote clusters in the *IBM Storage Scale Container Native* documentation.

3. Delete the secret that contains the remote Scale cluster user details.
   Command example:

   ```
   oc delete secret SCALE_CONNECTION_NAME-user-details-secret
   ```

Results

The remote IBM Storage Scale (CNSA) cluster is removed.

Configuring with IBM Fusion Access for Storage Area Network

Configure IBM Fusion Access for Storage Area Network (SAN) by connecting external SAN storage to Fusion Data Foundation from the Red Hat® OpenShift® Container Platform web console.

Before you begin

- You can configure IBM Fusion Access for SAN in Fusion Data Foundation as an external system only if Global Data Platform and Content-Aware Storage services are not installed. These services are mutually exclusive. Before planning or deploying IBM Fusion Access for SAN, ensure that your environment does not include Global Data Platform or Content-Aware Storage services.
- Install the Fusion Data Foundation version 4.21 or later with device type set to External by following the instructions mentioned in Installing Fusion Data Foundation.
- If you plan to use OpenShift internal image registry, ensure that the image registry is backed by a Persistent Volume Claim (PVC) and not an empty directory.
- Install the Kernel Module Management operator. To install it, do as follows:
  1. Log in to the Red Hat OpenShift Container Platform web console with administrative privileges.
  2. From the navigation menu, go to Ecosystem > Software Catalog.
  3. Select the Kernel Module Management operator and install it.
  4. Verify that the Kernel Module Management operator appears on the Installed Operators page.

Procedure

1. In the Red Hat OpenShift Container Platform web console, navigate to Storage > External systems.
2. Click Connect external systems.
3. Select Storage Area Network and click Next.
4. On the Connect Storage Area Network page, complete the General configuration:
   a. If you are using OpenShift external image registry, configure the following fields:
   Important: These fields are not visible or applicable when the OpenShift image registry is using persistent storage.
      i. Image registry URL: Specify the URL of the external image registry.
      ii. Image repository name: Enter the image repository name in the `<repository-name>/<image-name>` format.
      iii. Secret key: Select a secret used for image registry authentication.
   b. If Secure Boot is enabled, configure the following fields:
      i. CA certificate secret: Select a secret that contains the CA certificate.
      ii. Private key secret: Select a secret that contains the private key.
   c. Under Select local cluster nodes, choose one of the following:
      i. All nodes (Default): Uses all non-control-plane nodes to handle requests to IBM Scale.
      ii. Select nodes: Select a minimum of 3 nodes to handle requests to IBM Scale.
5. In the LUN group details section, complete the following configuration:
   a. In the Name field, enter a name for the LUN group.
   Note: A maximum of seven LUN groups is supported.
   b. Select one or more disks for the LUN group under Disks section.
   Note:
   A maximum of 128 disks can be added to a single LUN group.
6. Click Connect and create to complete the configuration.
   Note: To exit without saving changes, click Cancel.
7. Verify the IBM Fusion Access for SAN configuration:
   a. On the External systems page, click SAN Storage.
   b. On the SAN connection details page, verify:
      i. The Operator status shows a green check mark, indicating that the operator is running successfully.
      ii. The Connection status shows a green check mark, indicating that the SAN connection is established.
      iii. The expected number of LUN groups is displayed.
      iv. Each LUN group shows a Status of OK.
   c. From the navigation menu, go to StorageClasses.
   d. Verify that each LUN group has separate `StorageClasses` for container and VM workloads.
   For example, a LUN group named `lun-group-a` typically has:
      - `san-lun-group-a`: For general container storage
      - `san-lun-group-a-vm`: For VM workloads only

   Remember: When creating a virtual machine, ensure that you select the StorageClass with the `-vm` suffix.

Results

IBM Fusion Access for SAN is integrated with Fusion Data Foundation.

What to do next

If required, you can add more LUN groups to the configured IBM Fusion Access for SAN by following the instructions mentioned in Adding LUN groups to an IBM Fusion Access for SAN cluster.

- **Adding LUN groups to an IBM Fusion Access for SAN cluster**
  Provides steps to add additional LUN groups to an existing IBM Fusion Access for Storage Area Network (SAN) cluster using the Red Hat OpenShift Container Platform web console.

Adding LUN groups to an IBM Fusion Access for SAN cluster

Provides steps to add additional LUN groups to an existing IBM Fusion Access for Storage Area Network (SAN) cluster using the Red Hat® OpenShift® Container Platform web console.

Before you begin

Ensure that the following prerequisites are met:

- An IBM Fusion Access for SAN cluster is already configured as an external system. For instructions, see Configuring with IBM Fusion Access for Storage Area Network.
- You have administrative access to the Red Hat OpenShift Container Platform web console.

Procedure

1. Log in to the Red Hat OpenShift Container Platform web console with administrative privileges.
2. From the navigation menu, go to Storage > External systems.
3. On the External systems page, locate the configured SAN Storage cluster.
4. Click the More actions (⋮) icon at the end of the SAN Storage cluster row.
5. Select Add LUN group.
6. In the Create LUN group dialog, do the following steps:
   a. In the LUN group name field, enter a unique name for the new LUN group.
   b. Under Disks, select one or more available LUNs to include in the LUN group.
   c. Review the Path, WWID, and Capacity values for each selected disk to ensure correctness.
7. Click Connect and create to add the LUN group.
   Note: To exit without adding the LUN group, click Cancel.
8. Verify that the newly added LUN group appears under the IBM Fusion Access for SAN configuration:
   a. On the External systems page, click SAN Storage.
   b. On the SAN connection details page, verify:
      i. The newly added LUN group is displayed.
      ii. The newly added LUN group shows a Status of OK.
   c. From the navigation menu, go to StorageClasses.
   d. Verify that each LUN group has separate `StorageClasses` for container and VM workloads.
   For example, a LUN group named `lun-group-a` typically has:
   - `san-lun-group-a`: For general container storage
   - `san-lun-group-a-vm`: For VM workloads only

   Remember: When creating a virtual machine, ensure that you select the StorageClass with the `-vm` suffix.
9. Repeat steps 4 through 8 to add additional LUN groups as required.

Configuring Fusion Data Foundation in provider mode

Configure Fusion Data Foundation in provider mode that acts as the provider and base storage for the host cluster.

Before you begin

- Fusion Data Foundation service is installed with device type set to Local, as instructed in Installing Fusion Data Foundation.
- Do not use GPU nodes to configure Fusion Data Foundation, as these nodes are dedicated for AI workloads.

Procedure

1. On the Local storage page of the IBM Fusion web console, click Get started to initiate configuration of the Fusion Data Foundation service.
   It redirects you to the Create StorageSystem page in the Red Hat® OpenShift® Container Platform web console.
   Note: Ensure that you are logged in to the Red Hat OpenShift Container Platform web console as an administrator.
2. In the Backing storage page, select the Create a new StorageClass using the local storage devices option and click Next.
3. In the Advanced Settings page, you can configure the following optional items:
   - Enable Network Files System

     Automatically enable NFS for the cluster
   - Use Ceph RBD as the default StorageClass

     Sets Ceph RBD as the default StorageClass, eliminating the need to manually annotate a StorageClass.
   - Set default StorageClass for virtualization

     Marks the RBD virtualization storage class as the default for KubeVirt VM disks (PVs) after installation.
   - Use external PostgreSQL [Technology preview]

     Enables the use of an external PostgreSQL instance. This provides a high-availability solution for the Multicloud Object Gateway, where the PostgreSQL pod is a single point of failure.

     Important:
     Fusion Data Foundation ships PostgreSQL images maintained by IBM, which are used to store metadata for the Multicloud Object Gateway. This PostgreSQL usage is at the application level.

     As a result, Fusion Data Foundation does not perform database-level optimizations or in-depth insights.

     If you have well-maintained and optimized PostgreSQL instance, then it is recommended to use it. Fusion Data Foundation supports external PostgreSQL instances.

Any PostgreSQL-related issues requiring code changes or deep technical analysis may need to be addressed upstream. This could result in longer resolution times.

- Provide the following connection details:
  - Username
  - Password
  - Server name and Port
  - Database name
- Select Enable TLS/SSL checkbox to enable encryption for the Postgres server.

- Enable automatic backup

  Allows automatic backup for the Multicloud Object Gateway metadata database.

  - Select the backup frequency: Daily, Weekly, or Monthly.
  - Specify the number of backups to retain.

    Note: This option is disabled if the Use external PostgreSQL option is selected.
  - Click Next.

4. On the Create local volume set tab, do as follows:
   a. Enter a name for the LocalVolumeSet and the StorageClass.
   The local volume set name appears as the default value for the storage class name. You can change the name.
   Remember: The storage class name must not exceed 40 characters and must not match any of the following predefined storage classes:
   - `ocs-storagecluster-cephfs`
   - `ocs-storagecluster-ceph-rbd`
   - `ocs-storagecluster-ceph-rgw`
   - `openshift-storage.noobaa.io`
   - `ocs-storagecluster-ceph-rbd-virtualization`

   b. Select one of the following options:
   - Disks on all nodes

     Uses the available disks that match the selected filters on all the nodes.

   - Disks on selected nodes

     Uses the available disks that match the selected filters only on the selected nodes.

   Important:
   - The flexible scaling feature is enabled only when the storage cluster that you created with three or more nodes are spread across fewer than the minimum requirement of three availability zones.

     For information about flexible scaling, see knowledgebase article on *Scaling OpenShift Data Foundation cluster using YAML when flexible scaling is enabled*.

   - Flexible scaling features get enabled at the time of deployment and cannot be enabled or disabled later on.
   - If the nodes selected do not match the Fusion Data Foundation cluster requirement of an aggregated 30 CPUs and 72 GiB of RAM, a minimal cluster is deployed.

     For minimum starting node requirements, see Resource requirements.

   c. From the available list of Disk Type, select **`SSD/NVMe`**.
   Important: Ensure that you cleaned the disk before selecting it. For instructions, see Red Hat knowledge base article Uninstalling OpenShift Data Foundation in Internal mode.
   d. Expand the Advanced section and set the following options:
   - Volume Mode: Block is selected as the default value.
   - Device Type: Select one or more device types from the drop-down list
   - Disk Size: Set a minimum size of 100 GB for the device and maximum available size of the device that needs to be included.
   - Maximum Disks Limit: This indicates the maximum number of Persistent Volumes (PVs) that you can create on a node. If this field is left empty, then PVs are created for all the available disks on the matching nodes

   e. Click Next.
   A pop-up appears to confirm the creation of the LocalVolumeSet.
   f. Click Yes to continue.

5. On the Capacity and nodes tab, do as follows:
   a. Check the Available raw capacity.
   Available raw capacity is populated with the capacity value based on all the attached disks that are associated with the storage class. This takes some time to show up. The Selected nodes list shows the nodes based on the storage class.
   b. In the Configure performance tab, select one of the following resource profiles:
   Important: Before selecting a resource profile, ensure to check the current availability of resources within the cluster. Opting for a higher resource profile in a cluster with insufficient resources might lead to installation failures.
   - Lean

     Use this in a resource constrained environment with minimum resources that are lower than the recommended. This profile minimizes resource consumption by allocating fewer CPUs and less memory.

   - Balanced (default)

     Use this when recommended resources are available. This profile provides a balance between resource consumption and performance for diverse workloads.

   - Performance

     Use this in an environment with sufficient resources to get the best performance. This profile is tailored for high performance by allocating ample memory and CPUs to ensure optimal execution of demanding workloads.

For more information about resource requirements, see Resource requirements for performance profiles.

Note: You can configure the resource profile even after the deployment by selecting the Configure performance from the options menu in the StorageSystems tab.

c. Optional: Select the Taint nodes checkbox to dedicate the selected nodes for Fusion Data Foundation.
d. Click Next.

6. On the Security and network tab, configure the following based on your requirement:
   a. Select Enable data encryption for block and file storage to enable encryption.
   b. Select one or both of the following Encryption level:
      - **Cluster-wide encryption**

        Encrypts the entire cluster (block and file).

      - **StorageClass encryption**

        Creates encrypted persistent volume (block only) using encryption-enabled storage class.

   c. Optional: Select the Connect to an external key management service checkbox.
      This is optional for cluster-wide encryption.

      i. From the Key Management Service Provider drop-down list, either select Vault or Thales CipherTrust Manager (using KMIP).

      If you selected Vault, do to the next step. If you selected Thales CipherTrust Manager (using KMIP), do step 6.c.iii.

      ii. Select either of the following Authentication Method:

      - **Using Token authentication method**
        - Enter a unique Connection Name, host Address of the Vault server ('https://<hostname or ip>'), Port number, and Token.
        - Expand Advanced Settings to enter additional settings and certificate details based on your `Vault` configuration:
          - Enter the Key Value secret path in Backend Path that is dedicated and unique to Fusion Data Foundation.
          - Optional: Enter TLS Server Name and Vault Enterprise Namespace.
          - Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate, and Client Private Key.
        - Click Save and skip to step 6.d.
      - **Using Kubernetes authentication method**
        - Enter a unique Connection Name, host Address of the Vault server ('https://<hostname or ip>'), Port number, and Role name.
        - Expand Advanced Settings to enter additional settings and certificate details based on your `Vault` configuration:
          - Enter the Key Value secret path in Backend Path that is dedicated and unique to Fusion Data Foundation.
          - Optional: Enter TLS Server Name and Vault Enterprise Namespace.
          - Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate, and Client Private Key.
        - Click Save and skip to step 6.d.

      iii. To use **Thales CipherTrust Manager (using KMIP)** as the KMS provider, do the following steps:

      1. Enter a unique Connection Name for the Key Management service within the project.
      2. In the Address and Port sections, enter the IP of Thales CipherTrust Manager and the port where the KMIP interface is enabled. For example:
         - Address: 123.34.3.2
         - Port: 5696
      3. Upload the Client Certificate, CA certificate, and Client Private Key.
      4. If StorageClass encryption is enabled, enter the Unique Identifier to be used for encryption and decryption that is generated in the previous step.
      5. The TLS Server field is optional and used when there is no DNS entry for the KMIP endpoint. For example, `kmip_all_<port>.ciphertrustmanager.local`.

   d. Select Host as the Network option.
   e. Click Next.
7. On the Review and create tab, review the configuration details.
   To modify any configuration settings, click Back to go back to the previous configuration page.
8. Click Create StorageSystem.
9. Create a load balancer service in the `openshift-storage` namespace as follows:
   To create a load balancer service in the `openshift-storage` namespace, you need the IP address pool name from the MetalLB operator. If the MetalLB operator is already installed and configured in your cluster, go to step 9.d. If not, complete steps 9.a, 9.b, and 9.c first.
   Note: Steps 9.a, 9.b, and 9.b are applicable starting from FDF 4.19 onward.
   a. From the Red Hat Operator Catalog, install MetalLB 4.16 or higher.
      Add the MetalLB operator to your cluster so that when a Service of type `LoadBalancer` is added to the cluster, MetalLB can add a fault-tolerant external IP address to the Service. For the procedure to install and validate MetalLB, see Red Hat Documentation. Go to your specific version of OpenShift Container Platform and check the MetalLB details.
   b. Create a MetalLB CR.
      For example:

      ```
      apiVersion: metallb.io/v1beta1
      kind: MetalLB
      metadata:
        name: metallb
        namespace: metallb-system
      ```

   c. Create an IP address pool.
      Note: Reserve a set of unused IPs from the same CIDR as the Bare Metal network for MetalLB. MetalLB uses these IPs for any load balancer service created on the cluster, and not only for the Hosted Control Plane.
      For example:

      ```
      apiVersion: metallb.io/v1beta1
      kind: IPAddressPool
      metadata:
        name: metallb
        namespace: metallb-system
      ```

```
spec:
  addresses:
  - 9.9.0.51-9.9.0.70
```

Note: When you configure the load balancer, other applications can also use the advertised addresses. Ensure that you have enough addresses for all workloads on this cluster and for any other OpenShift Container Platform clusters that you create.

d. Define a load balancer service YAML.
For example:

```
kind: Service
apiVersion: v1
metadata:
  name: ocs-provider-server-load-balancer
  namespace: openshift-storage
  annotations:
    metallb.universe.tf/ip-allocated-from-pool: <address-pool-name-from-metallb>
spec:
  ports:
    - name: provider
      protocol: TCP
      port: 50051
      targetPort: ocs-provider
      nodePort: 30756
  type: LoadBalancer
  selector:
    app: ocsProviderApiServer
```

Remember: Replace the **<address-pool-name-from-metallb>** with the IP address pool name configured in the MetalLB operator. For example, **metallb** is the IP address pool name used in the example provided step 9.c.

e. Apply the YAML.
Command example:

```
oc apply -f ocs-provider-server-load-balancer.yaml
```

10. Validate whether Fusion Data Foundation is successfully configured in provider mode by following the instructions that are mentioned in Verifying Fusion Data Foundation provider mode configuration.

What to do next

- To add new nodes to increase storage capacity in the cluster, see Scaling out storage capacity.
- To add new disks to a storage node, see Scaling up storage capacity.

- **Verifying Fusion Data Foundation provider mode configuration**
  Validate whether Fusion Data Foundation is successfully configured in provider mode.

Verifying Fusion Data Foundation provider mode configuration

Validate whether Fusion Data Foundation is successfully configured in provider mode.

Procedure

1. Verify the final status of the installed storage cluster.
   a. Go to Storage > Data Foundation > Storage System > ocs-storagecluster.
   b. Verify that the **Status** of the **StorageCluster** is **Ready** and has a green tick mark next to it.
2. In the IBM Fusion web console, go to Storage > Local Storage page and check the Fusion Data Foundation health status.
   If the Fusion Data Foundation cluster is not displayed on the Local storage page or the following error message appears, the issue might be related to **deviceType**:

   ```
   No worker nodes The Local Storage Operator was unable to find the candidate storage nodes
   ```

   To resole the issue, do the following steps:

   a. Check whether the **deviceType** value exists in the storagecluster CR:

      Command example:

      ```
      oc get storageclusters.ocs.openshift.io -n openshift-storage ocs-storagecluster -o
      jsonpath='{.spec.storageDeviceSets[0].deviceType}'
      ```

   b. If the **deviceType** is empty or incorrect, update the **storagecluster.storageDeviceSets.deviceType** field. Use **SSD** when the storage cluster uses SSD or NVMe drives, and use **HDD** when it uses HDD drives.

      ```
      oc edit storageclusters.ocs.openshift.io -n openshift-storage ocs-storagecluster
      ```

      For example:

      ```
      storageDeviceSets:
        - config: {}
          count: 3
          dataPVCTemplate:
            metadata: {}
            spec:
              accessModes:
              - ReadWriteOnce
              resources:
      ```

```
          requests:
            storage: "1"
        storageClassName: ibm-spectrum-fusion-local
        volumeMode: Block
      status: {}
    deviceType: HDD
    name: ocs-deviceset-ibm-spectrum-fusion-local
    placement: {}
    preparePlacement: {}
    replica: 1
    resources: {}
```

Configuring Fusion Data Foundation in MCG-only mode

Configure Fusion Data Foundation in Multicloud Object Gateway (MCG)-only mode to provide scalable object storage without block or file services.

Before you begin

- Fusion Data Foundation is installed in dynamic mode, as instructed in Installing Fusion Data Foundation.
- You are logged in to the Red Hat® OpenShift® Container Platform web console as an administrator.
- Project is set to **openshift-storage**.

About this task

Fusion Data Foundation can be deployed in MCG-only mode, where only the MCG service is installed. In this mode, the system provides object storage exclusively and does not support block or file storage.

Procedure

1. On the Red Hat OpenShift Container Platform web console, go to Storage > Storage Cluster > Configure Data Foundation.
2. In the Welcome to Data Foundation dialog, click the Setup Multicloud Object Gateway tile.
3. On the Backing storage page, select Use an existing StorageClass or Create a new StorageClass for the Backing storage type.
4. Click Next.
5. On the Review and create page, review the configuration details and click Create StorageSystem.
   Tip: To modify any configuration details, click Back.
6. Do the following verification steps:
   a. Verify that the StorageSystem resource is created.
   Command example:

   ```
   oc get StorageSystem -n openshift-storage
   ```

   Example output:

   ```
   NAME                                 STORAGE-SYSTEM-KIND                  STORAGE-SYSTEM-NAME
   ocs-storagecluster-storagesystem   storagecluster.ocs.openshift.io/v1   ocs-storagecluster
   ```

   b. Verify that the PersistentVolumeClaim (PVC) is created as the backing storage.
   Command example:

   ```
   oc get pvc -n openshift-storage
   ```

   Example output:

   ```
   NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS          V
   db-noobaa-db-pg-0   Bound    pvc-24fb7c6d-2964-4609-859e-5468d6f20461   50Gi       RWO            fusion-gdp-sc-cp-sc   <
   ```

   c. Verify that the StorageCluster is in the **Ready** phase.
   Command example:

   ```
   oc get StorageCluster -n openshift-storage
   ```

   Example output:

   ```
   NAME                 AGE     PHASE   EXTERNAL   CREATED AT             VERSION
   ocs-storagecluster   2m59s   Ready              2025-08-08T19:18:46Z   4.17.9
   ```

   d. Verify that the noobaa Pods are in **Running** status.
   Command example:

   ```
   oc get pod -n openshift-storage
   ```

   Example output:

   ```
   NAME                                               READY   STATUS    RESTARTS   AGE
   ceph-csi-controller-manager-6b45485f4f-hlv7j       2/2     Running   0          3h58m
   csi-addons-controller-manager-867f5596fd-ftfzk     2/2     Running   0          3h58m
   noobaa-core-0                                      2/2     Running   0          3m27s
   noobaa-db-pg-0                                     1/1     Running   0          3m28s
   noobaa-default-backing-store-noobaa-pod-5a57efda   1/1     Running   0          79s
   noobaa-endpoint-d7f495757-9v5vp                    1/1     Running   0          104s
   noobaa-operator-6b88f87cf6-phcgx                   1/1     Running   0          3h58m
   ocs-metrics-exporter-649f675dd9-vksg5              1/1     Running   0          3m27s
   ocs-operator-554858f998-pkp4t                      1/1     Running   0          3h57m
   odf-console-656656598-djnz7                        1/1     Running   0          3h58m
   ```

```
odf-operator-controller-manager-77ff48499c-nptnh   2/2     Running   0          3h58m
rook-ceph-operator-66ccffcc44-skbrf                1/1     Running   0          3h57m
ux-backend-server-6b4d8fb86f-r97qn                 2/2     Running   0          3h57m
```

e. Verify that the noobaa resources are in the **Ready** phase.
Command example:

```
oc get -n openshift-storage noobaas.noobaa.io,bucketclasses.noobaa.io,noobaas.noobaa.io
```

Example output:

```
NAME                       S3-ENDPOINTS                     STS-ENDPOINTS                     SYSLOG-ENDPOINTS    IMAGE
noobaa.noobaa.io/noobaa    ["https://9.11.110.227:31135"]   ["https://9.11.110.227:30345"]                       registry.r

NAME                                                TYPE            PHASE   AGE
backingstore.noobaa.io/noobaa-default-backing-store   s3-compatible   Ready   9m17s

NAME                                                PLACEMENT                                                                NAME
bucketclass.noobaa.io/noobaa-default-bucket-class   {"tiers":[{"backingStores":["noobaa-default-backing-store"]}]}
```

f. In the Red Hat OpenShift Container Platform web console, go to Storage > Data Foundation and check that the storage system is healthy.
g. In the IBM Fusion web console, go to Storage > Local storage and check that the Multi-cloud Gateway mode is detected.

Results

Fusion Data Foundation is configured and running in MCG-only mode.

Scaling out storage capacity

Fusion Data Foundation is highly scalable and can be expanded by adding new nodes with sufficient storage, CPU, and RAM resources.

Add nodes to increase storage capacity when existing worker nodes reached the maximum supported OSDs or when resources are insufficient to add new OSDs on existing nodes. There is no limit on the number of nodes that can be added. However, from a technical support perspective, the limit for Fusion Data Foundation is 2000 nodes.

It is recommended to add nodes in multiples of three, placing each node in a different failure domain or zone. Repeat this procedure for each node. For bare metal installer-provisioned infrastructure, expand the cluster before proceeding.

While adding nodes in multiples of three is recommended, you can still add one node at a time in a flexible scaling deployment. To verify whether flexible scaling is enabled, see Red Hat® Knowledgebase article Scaling OpenShift Data Foundation cluster using YAML when flexible scaling is enabled.

Important: Fusion Data Foundation does not support heterogeneous disk size and types. Add only nodes with disks that match the type and size that is used during the initial deployment.
To scale out storage capacity, do as follows:

1. Add new nodes.
2. Scale up the storage capacity.

The procedure for adding nodes varies based on the type of infrastructure, as follows:

- **Installer-provisioned infrastructure**: See Adding a node to an installer-provisioned infrastructure.
- **User-provisioned infrastructure**: See Adding a node to a user-provisioned infrastructure.

- **Adding a node to an installer-provisioned infrastructure**
  Add nodes to increase storage capacity for Fusion Data Foundation to an installer-provisioned infrastructure when existing worker nodes reached the maximum supported OSDs or when resources are insufficient to add new OSDs on existing nodes.
- **Adding a node to a user-provisioned infrastructure**
  Add nodes to increase storage capacity for Fusion Data Foundation to a user-provisioned infrastructure when existing worker nodes reached the maximum supported OSDs or when resources are insufficient to add new OSDs on existing nodes.

Adding a node to an installer-provisioned infrastructure

Add nodes to increase storage capacity for Fusion Data Foundation to an installer-provisioned infrastructure when existing worker nodes reached the maximum supported OSDs or when resources are insufficient to add new OSDs on existing nodes.

Before you begin

- You are logged in to the Red Hat® OpenShift® Container Platform web console as an administrator.
- Fusion Data Foundation service is installed and configured.
- It is recommended to add nodes in multiples of three, placing each node in a different failure domain or zone. Repeat this procedure for each node. For bare metal installer-provisioned infrastructure, expand the cluster before proceeding.

  While adding nodes in multiples of three is recommended, you can still add one node at a time in a flexible scaling deployment. To verify whether flexible scaling is enabled, see Red Hat Knowledgebase article Scaling OpenShift Data Foundation cluster using YAML when flexible scaling is enabled.

  Important: Fusion Data Foundation does not support heterogeneous disk size and types. Add only nodes with disks that match the type and size that is used during the initial deployment.

Procedure

1. Go to Compute > MachineSets.
2. On the MachineSets page, select Edit Machine Count from the Action Menu (⋮) of the machine set where you want to add nodes.
3. Specify the number of nodes to add, and click Save.
4. Click Compute > Nodes and verify that the new node is in Ready state.
5. Apply the Fusion Data Foundation label to the new node.x
    a. Click the Action Menu (⋮) for the new node and select Edit labels.
    b. Add the `cluster.ocs.openshift.io/openshift-storage` label and click Save.
6. Do the following verification steps:
    a. Run the following command and verify that the new node is present in the command output:
    Command example:

    ```
    oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
    ```

    Output example:

    ```
    r8-ru10-w-1.stg-odf-211.fusion.tadn.ibm.com
    r8-ru12-w-1.stg-odf-211.fusion.tadn.ibm.com
    r8-ru14-w-1.stg-odf-211.fusion.tadn.ibm.com
    r8-ru8-w-1.stg-odf-211.fusion.tadn.ibm.com
    ```

    b. Go to Workloads > Pods and check that the following Pods on the new node are in the `Running` state:
        - `openshift-storage.cephfs.csi.ceph.com-*`
        - `openshift-storage.rbd.csi.ceph.com-*`

Results

Additional nodes are added to the infrastructure and are available for use in the Fusion Data Foundation cluster.

What to do next

Scale up the storage capacity by following the instructions mentioned in Scaling up storage capacity.

Adding a node to a user-provisioned infrastructure

Add nodes to increase storage capacity for Fusion Data Foundation to a user-provisioned infrastructure when existing worker nodes reached the maximum supported OSDs or when resources are insufficient to add new OSDs on existing nodes.

Before you begin

- You are logged in to the Red Hat® OpenShift® Container Platform web console as an administrator.
- Fusion Data Foundation service is installed and configured.
- It is recommended to add nodes in multiples of three, placing each node in a different failure domain or zone. Repeat this procedure for each node. For bare metal installer-provisioned infrastructure, expand the cluster before proceeding.

  While adding nodes in multiples of three is recommended, you can still add one node at a time in a flexible scaling deployment. To verify whether flexible scaling is enabled, see Red Hat Knowledgebase article Scaling OpenShift Data Foundation cluster using YAML when flexible scaling is enabled.

  Important: Fusion Data Foundation does not support heterogeneous disk size and types. Add only nodes with disks that match the type and size that is used during the initial deployment.

Procedure

1. Based on your infrastructure type, complete these steps:
    a. Provision a new machine that meets the required specifications.
    For more information on infrastructure requirements, see Platform requirements.
    b. Use the new machine to create a new Red Hat OpenShift Container Platform worker node.
2. Check for certificate signing requests (CSRs) that are in `Pending` state.
    Command example:

    ```
    oc get csr
    ```

3. Approve all the required CSRs for the new node.
    Command example:

    ```
    oc adm certificate approve <Certificate_Name>
    ```

4. Click Compute > Nodes and verify that the new node is in Ready state.
5. Apply the Fusion Data Foundation label to the new node by using any one of the following methods:

    User interface
        a. Click the Action Menu (⋮) for the new node and select Edit labels.
        b. Add the `cluster.ocs.openshift.io/openshift-storage` label and click Save.

    Command line interface
        Run the following `oc` command on the new node.

```
oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

6. In the Red Hat OpenShift Container Platform web console, go to Operators > Installed Operators.
   In the Project drop-down list, ensure that you selected the required project.
7. Click Local Storage.
8. Select the Local Volume Discovery tab and do as follows:
   a. Click the Action menu (⋮) next to **LocalVolumeDiscovery** and select Edit LocalVolumeDiscovery.
   b. In the YAML tab, add the hostname of the new node in the **values** field under **nodeSelector**.
   c. Click Save.
9. Select the Local Volume Sets tab and do as follows:
   a. Click the Action menu (⋮) next to **LocalVolumeSet** and select Edit LocalVolumeSet.
   b. In the YAML tab, add the hostname of the new node in the **values** field under **nodeSelector**.
   c. Click Save.
10. Do the following verification steps:
    a. Run the following command and verify that the new node is present in the command output:
       Command example:

       ```
       oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
       ```

       Output example:

       ```
       r8-ru10-w-1.stg-odf-211.fusion.tadn.ibm.com
       r8-ru12-w-1.stg-odf-211.fusion.tadn.ibm.com
       r8-ru14-w-1.stg-odf-211.fusion.tadn.ibm.com
       r8-ru8-w-1.stg-odf-211.fusion.tadn.ibm.com
       ```

    b. Go to Workloads > Pods and check that the following Pods on the new node are in the **Running** state:
       - **openshift-storage.cephfs.csi.ceph.com-***
       - **openshift-storage.rbd.csi.ceph.com-***

Results

Additional nodes are added to the infrastructure and are available for use in the Fusion Data Foundation cluster.

What to do next

Scale up the storage capacity by following the instructions mentioned in Scaling up storage capacity.

Scaling up storage capacity

Add a new disk to a storage node to increase the storage capacity of the Fusion Data Foundation cluster.

Before you begin

- You are logged in to the Red Hat® OpenShift® Container Platform web console as an administrator.
- Fusion Data Foundation service is installed and configured.
- The new disks must be the same size as the disks used during deployment because Fusion Data Foundation does not support heterogeneous disks or OSDs.

  For deployments with three failure domains, scale up storage by adding disks in multiples of three, ensuring that the same number of disks are added to nodes in each failure domain. For example, if you add six disks, two disks must come from nodes in each of the three failure domains. If the number of disks is not a multiple of three, the system uses disks up to the nearest multiple of three, and the remaining disks remain unused.

  For deployments with lesser than three failure domains, you can add any number of disks. Ensure that flexible scaling is enabled. To verify whether flexible scaling is enabled, see Red Hat Knowledgebase article Scaling OpenShift Data Foundation cluster using YAML when flexible scaling is enabled.

  Note: The flexible scaling feature is enabled at the time of Fusion Data Foundation deployment and cannot be enabled or disabled later.
- Ensure that the disks for scaling are attached to the storage node.
- Ensure that the **LocalVolumeDiscovery** and **LocalVolumeSet** objects are created.

Procedure

1. Go to Storage > Data Foundation.
2. Select the Storage Systems tab and do as follows:
   a. Click the Action menu (⋮) next to the storage cluster where you want to add capacity and select Add Capacity.
   b. Select either an existing StorageClass (where you added new disks) or a new StorageClass, depending on your requirement.
      The Available capacity displayed is based on the local disks available in the storage class.
   c. Click Add.
3. To check the status, go to Storage > Data Foundation and verify that the storage cluster is in the Ready status.
4. Verify the Raw Capacity card.
   a. Go to Storage > Data Foundation.
   b. Select the Storage Systems tab.
   c. Click the storage cluster link to view its details.
   d. Go to Overview > Block and File tab, check the Raw Capacity pane.
      The capacity increases based on your selections.
      Note: The raw capacity does not consider replication and shows the full capacity.
5. Verify that the new OSDs and their corresponding new PersistentVolumeClaims (PVCs) are created.
   a. To view the state of the newly created OSDs, go to Workloads > Pods.

From the Project drop-down list, ensure to select **openshift-storage**.
Note: If the Show default projects option is disabled, enable it using the toggle button to display all default projects.

b. To view the state of the PVCs, go to Storage > Persistent Volume Claims.
From the Project drop-down list, ensure to select **openshift-storage**.
Note: If the Show default projects option is disabled, enable it using the toggle button to display all default projects.

6. Optional: If cluster-wide encryption is enabled, verify that the new OSD devices are encrypted.
   a. Identify the nodes where the new OSD pods are running.
   Command example:

   ```
   oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
   ```

   Example output:

   ```
   NODE
   compute-1
   ```

   b. For each node identified in the previous step, do the following steps:
      i. Create a debug pod and open a chroot environment for the selected hosts.

      Command example:

      ```
      oc debug node/<node-name>
      ```

      ```
      chroot /host
      ```

      ii. Check for the **crypt** keyword next to the **ocs-deviceset** names.

      Command example:

      ```
      lsblk
      ```

Results

The new disks are detected and added to the storage cluster.

Enabling automatic capacity scaling

Enable automatic capacity scaling on the Fusion Data Foundation cluster configured with dynamic storage devices to ensure seamless scaling to meet demand.

Before you begin

- You are logged in to the Red Hat® OpenShift® Container Platform web console as an administrator.
- Fusion Data Foundation service is installed and configured.

Important: IBM Cloud 10 IOPS-tier workloads are not supported.

About this task

When automatic capacity scaling is enabled, additional raw capacity equivalent to the configured deployment size is automatically added to the cluster when the used capacity reaches 70%. This ensures your Fusion Data Foundation cluster scales seamlessly to meet demand.

This option is disabled in lean profile mode, local mode, and external mode deployment.

Important: Enabling automatic capacity scaling may incur additional costs for the underlying storage.

Procedure

1. Go to Storage > Data Foundation.
2. Click the Storage Systems tab.
3. Click the Action menu (⋮) next to the storage cluster where you want to enable automatic capacity scaling.
4. Select Automatic capacity scaling.
5. On the Automatic capacity scaling page, select the Enable automatic capacity scaling for your cluster checkbox.
6. Set the cluster expansion limit from the drop-down list.
   This defines the maximum capacity that the cluster can expand to in the cloud. Automatic scaling is suspended once this limit is reached.
7. Click Save changes.

Results

Automatic capacity scaling is enabled.

Preparing to connect to an external KMS server in Fusion Data Foundation

Procedure to prepare for the connection to an external KMS from Fusion Data Foundation.

Before you begin

- For external Key Management System (KMS), choose either HashiCorp Vault or Thales CipherTrust Manager.
- You must install Fusion Data Foundation from the Services page of the user interface and ensure that it is in running state.
  For the procedure to install, see Data Foundation.
- For HashiCorp Vault, select an unique path name as the backend path that follows the naming convention. If you change this path name later, then the data becomes inaccessible.
- For Thales CipherTrust Manager, enable the Key Management Interoperability Protocol.
- Ensure that you are using signed certificates on your KMS servers.

About this task

This procedure is used in Configuring Fusion Data Foundation in local mode and Configuring Fusion Data Foundation in dynamic mode.

Fusion Data Foundation supports cluster-wide encryption (encryption-at-rest) for all the disks and Multicloud Object Gateway operations in the storage cluster. The keys are stored using a Kubernetes secret or an external KMS. When you store the keys by using a Kubernetes secret, no need for you to do manual steps. You can enable cluster-wide encryption when you deploy Fusion Data Foundation.

This procedure provides steps (manual part) to initialize encryption configuration with KMS before you enable encryption. For HashiCorp Vault, you can choose either the token authentication method or the Kubernetes authentication method.

If errors occur, see Fusion Data Foundation service error scenarios. For additional reference, go through the following documents:

- Security considerations
- Vault server documentation
- Thales CipherTrust Manager documentation

- **Enabling encryption with the token authentication using HashiCorp Vault(manual part)**
  Configure token settings in vault server.
- **Enabling encryption with the Kubernetes authentication using HashiCorp Vault (manual part)**
  Configure Kubernetes settings in vault server.
- **Enabling encryption using Thales CipherTrust Manager (manual part)**
  Configure the Key Management Interoperability Protocol (KMIP) settings in Thales CipherTrust Manager server.

Enabling encryption with the token authentication using HashiCorp Vault(manual part)

Configure token settings in vault server.

Procedure

1. Enable the Key/Value (KV) backend path in Vault. Run the following command for Vault KV secret engine API, version 2, as it supports only version 2.

   ```
   vault secrets enable -path=odf kv-v2
   ```

   This is a one time settings. You can get the token from vault administrator directly, or you can login vault server and create a token with following steps. Use an unique path name as the backend path that follows the naming convention. You cannot change it later.
   Note: The example uses backend path **odf**.
2. Create a policy with certain permission on the secret using the following commands.

   ```
   echo '
   path "odf/*" {
     capabilities = ["create", "read", "update", "delete", "list"]
   }
   path "sys/mounts" {
   capabilities = ["read"]
   }'| vault policy write odf -
   ```

3. Create a token matching the policy.

   ```
   vault token create -policy=odf -format json
   ```

   Example output:

   Take a note for the `auth.client_token`

   ```
   # vault token create -policy=odf -format json
   {
   "request_id": "e7103f23-c67c-2412-77b0-b92a728fd5fd",
   "lease_id": "",
   "lease_duration": 0,
   "renewable": false,
   "data": null,
   "warnings": null,
   "auth": {
       "client_token": "hvs.CAESILkFSEnBgMgE0JxdbnBf_VUPxlvhCPTBBdx30jlbMfiMGh4KHGh2cy5QeFZVWnhOWGMxdlhvSU5YU0tNRjBBVGo",
       "accessor": "IFNbUOwWAzg5cyiUZGVlvf5o",
       "policies": [
       "default",
       "odf"
   ```

```
    ],
    "token_policies": [
    "default",
    "odf"
    ],
    "identity_policies": null,
    "metadata": null,
    "orphan": false,
    "entity_id": "",
    "lease_duration": 2764800,
    "renewable": true,
    "mfa_requirement": null
}
}
```

Enabling encryption with the Kubernetes authentication using HashiCorp Vault (manual part)

Configure Kubernetes settings in vault server.

Procedure

1. Get the `VAULT_SA_SECRET_NAME` in OpenShift® Container Platform:
   a. For OpenShift Container Platform 4.14, identify the secret name associated with the `serviceaccount` (SA) created previously.

   ```
   VAULT_SA_SECRET_NAME=$(oc -n openshift-storage get sa odf-vault-auth -o jsonpath="{.secrets[*]['name']}" | grep -o "
   [^[:space:]]*-token-[^[:space:]]*")
   ```

   Example output:

   ```
   [root@fu40 ~]# echo $VAULT_SA_SECRET_NAME
   odf-vault-auth-token-8kb2r
   ```

   b. For OpenShift Container Platform 4.15, assign a default value for `VAULT_SA_SECRET_NAME`"

   ```
   VAULT_SA_SECRET_NAME=odf-vault-auth-token
   ```

2. Get the parameters from OpenShift Container Platform cluster.
   a. Get `SA_JWT_TOKEN` and `SA_CA_CRT` from the secret.

   ```
   SA_JWT_TOKEN=$(oc -n openshift-storage get secret "$VAULT_SA_SECRET_NAME" -o jsonpath="{.data.token}" | base64 --
   decode; echo)
   ```

   b. Get `SA_CA_CRT` from the secret.

   ```
   SA_CA_CRT=$(oc -n openshift-storage get secret "$VAULT_SA_SECRET_NAME" -o jsonpath="{.data['ca\.crt']}" | base64 --
   decode; echo)
   ```

   c. Retrieve the OpenShift Container Platform cluster endpoint `OCP_HOST`.

   ```
   OCP_HOST=$(oc config view --minify --flatten -o jsonpath="{.clusters[0].cluster.server}")
   ```

   d. Fetch the service account issuer:

   ```
   oc proxy &
   proxy_pid=$!
   issuer="$( curl --silent http://127.0.0.1:8001/.well-known/openid-configuration | jq -r .issuer)"
   kill $proxy_pid
   ```

   Example output:

   ```
   # oc proxy &
   # proxy_pid=$!
   # issuer="$( curl --silent http://127.0.0.1:8001/.well-known/openid-configuration | jq -r .issuer)"
   # echo $issuer
   https://kubernetes.default.svc
   #     kill $proxy_pid
   ```

3. Apply the settings in the vault server.
   a. Use the information collected in the previous section to set up the Kubernetes authentication method in the vault server.

   ```
   vault auth enable kubernetes

   vault write auth/kubernetes/config \
             token_reviewer_jwt="$SA_JWT_TOKEN" \
             kubernetes_host="$OCP_HOST" \
             kubernetes_ca_cert="$SA_CA_CRT" \
             issuer="$issuer"
   ```

   Example output:

   ```
   # vault write auth/kubernetes/config \
   >           token_reviewer_jwt="$SA_JWT_TOKEN" \
   >           kubernetes_host="$OCP_HOST" \
   >           kubernetes_ca_cert="$SA_CA_CRT" \
   >           issuer="$issuer"
   Success! Data written to: auth/kubernetes/config
   ```

b. Enable the Key/Value (KV) backend path in Vault. For the vault KV secret engine API, see version 2. Use a unique path name as the backend path. The name should not be changed, and it should be consistent with the value of Backend Path input that present in the user interface.

```
vault secrets enable -path=<repalce-with-the-backend-path> kv-v2
```

The example sample uses backend path odf.

```
vault secrets enable -path=odf kv-v2
```

c. Create a policy to restrict users to perform a write or delete operation on the secret using the following commands.

```
echo '
path "odf/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}
path "sys/mounts" {
capabilities = ["read"]
}'| vault policy write odf -
```

d. Generate the `odf-rook-ceph-op` role using the following commands.

```
vault write auth/kubernetes/role/odf-rook-ceph-op \
        bound_service_account_names=rook-ceph-system,rook-ceph-osd,noobaa \
        bound_service_account_namespaces=openshift-storage \
        policies=odf \
        ttl=1440h
```

Example output:

```
#     vault write auth/kubernetes/role/odf-rook-ceph-op \
>             bound_service_account_names=rook-ceph-system,rook-ceph-osd,noobaa \
>             bound_service_account_namespaces=openshift-storage \
>             policies=odf \
>             ttl=1440h
Success! Data written to: auth/kubernetes/role/odf-rook-ceph-op
```

The role `odf-rook-ceph-op` is previously input in the role field used when you configure the KMS connection details in the IBM Fusion user interface.

e. Generate the `odf-rook-ceph-osd` role using the following commands.

```
vault write auth/kubernetes/role/odf-rook-ceph-osd \
        bound_service_account_names=rook-ceph-osd \
        bound_service_account_namespaces=openshift-storage \
        policies=odf \
        ttl=1440h
```

Example output:

```
#  vault write auth/kubernetes/role/odf-rook-ceph-osd \
>         bound_service_account_names=rook-ceph-osd \
>         bound_service_account_namespaces=openshift-storage \
>         policies=odf \
>         ttl=1440h
Success! Data written to: auth/kubernetes/role/odf-rook-ceph-osd
```

- **Enabling key rotation for Kubernetes authentication**
  Enabling key rotation for Kubernetes authentication helps to achieve the security common practices that require periodic encryption key rotation.
- **Disabling key rotation for Kubernetes authentication**
  Key rotation for Kubernetes persistent volume claims (PVCs) can be disabled at the storage class level or for a specific PVC by updating the relevant annotations and patch settings.

Enabling key rotation for Kubernetes authentication

Enabling key rotation for Kubernetes authentication helps to achieve the security common practices that require periodic encryption key rotation.

To enable key rotation, add the annotation *keyrotation.csiaddons.openshift.io/schedule: <value>* to either Namespace, StorageClass, or PersistentVolumeClaim in order of precedence.

Set the *<value>* either *@hourly*, *@daily*, *@weekly*, *@monthly*, or *@yearly* based on your requirement. If the *<value>* is empty, *@weekly* is set by default.

Important: Only RADOS Block Device (RBD)-related volumes support key rotation.
Examples of enabling key rotation with the value *@weekly* are as follows:

Namespace

For example, run the following command to get the *default* namespace details:

```
 oc get namespace default
```

Example output:

```
NAME      STATUS   AGE
default   Active   5d2h
```

Run the following command to enable the key rotation to the default namespace:

```
$ oc annotate namespace default "keyrotation.csiaddons.openshift.io/schedule=@weekly"
```

Example output:

```
namespace/default annotated
```

StorageClass

For example, run the following command to get the *rbd-sc* StorageClass details:

```
oc get storageclass rbd-sc
```

Example output:

```
NAME       PROVISIONER         RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
rbd-sc     rbd.csi.ceph.com    Delete          Immediate           true                   5d2h
```

Run the following command to enable the key rotation to the *rbd-sc* StorageClass:

```
oc annotate storageclass rbd-sc "keyrotation.csiaddons.openshift.io/schedule=@weekly"
```

Example output:

```
storageclass.storage.k8s.io/rbd-sc annotated
```

PersistentVolumeClaim

For example, run the following command to get the *data-pvc* PersistentVolumeClaim details:

```
oc get pvc data-pvc
```

Example output:

```
NAME       STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS       AGE
data-pvc   Bound    pvc-f37b8582-4b04-4676-88dd-e1b95c6abf74   1Gi        RWO            default            20h
```

Run the following command to enable the key rotation to the *data-pvc* PersistentVolumeClaim:

```
oc annotate pvc data-pvc "keyrotation.csiaddons.openshift.io/schedule=@weekly"
```

Example output:

```
persistentvolumeclaim/data-pvc annotated
```

Replacing key rotation schedule

To replace the current key rotation schedule with a new one, follow these steps:

1. Run the following command to get the existing schedule details:

   ```
   oc get encryptionkeyrotationcronjobs.csiaddons.openshift.io
   ```

   Example output:

   ```
   NAME                      SCHEDULE     SUSPEND   ACTIVE   LASTSCHEDULE   AGE
   data-pvc-1642663516       @weekly                                        3s
   ```

2. Run the following command to overwrite the existing schedule with a new one:

   ```
   oc annotate pvc data-pvc "keyrotation.csiaddons.openshift.io/schedule=*/1 * * * *" --overwrite=true$
   ```

   Example output:

   ```
   persistentvolumeclaim/data-pvc annotated
   ```

3. Run the following command to check whether the new schedule is set:

   ```
   oc get encryptionkeyrotationcronjobs.csiaddons.openshift.io
   ```

   Example output:

   ```
   NAME                    SCHEDULE     SUSPEND   ACTIVE   LASTSCHEDULE   AGE
   data-pvc-1642664617     */1 * * * *                                    3s
   ```

Disabling key rotation for Kubernetes authentication

Key rotation for Kubernetes persistent volume claims (PVCs) can be disabled at the storage class level or for a specific PVC by updating the relevant annotations and patch settings.

You can disable key rotation for the following:

- All the persistent volume claims (PVCs) of storage class
- A specific PVC

Disabling key rotation for all PVCs of a storage class

To disable key rotation for all PVCs, update the annotation of the storage class:

```
oc get storageclass rbd-sc
NAME       PROVISIONER         RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
rbd-sc     rbd.csi.ceph.com    Delete          Immediate           true                   5d2h

oc annotate storageclass rbd-sc "keyrotation.csiaddons.openshift.io/enable: false"
storageclass.storage.k8s.io/rbd-sc annotated
```

Disabling key rotation for a specific persistent volume claim

1. Identify the **EncryptionKeyRotationCronJob** CR for the PVC you want to disable key rotation on:

   ```
   oc get encryptionkeyrotationcronjob -o jsonpath='{range .items[?(@.spec.jobTemplate.spec.target.persistentVolumeClaim=="
   <PVC_NAME>")]}{.metadata.name}{"\n"}{end}'
   ```

   Where **<PVC_NAME>** is the name of the PVC that you want to disable.
2. Apply the following to the **EncryptionKeyRotationCronJob**CR from the previous step to disable the key rotation:
   a. Update the **csiaddons.openshift.io/state** annotation from **managed** to **unmanaged**:

      ```
      oc annotate encryptionkeyrotationcronjob <encryptionkeyrotationcronjob_name> "csiaddons.openshift.io/state=unmanaged"
      --overwrite=true
      ```

      Where <encryptionkeyrotationcronjob_name> is the name of the **EncryptionKeyRotationCronJob** CR.
   b. Add **suspend: true** under the **spec** field:

      ```
      oc patch encryptionkeyrotationcronjob <encryptionkeyrotationcronjob_name> -p '{"spec": {"suspend": true}}' --
      type=merge.
      ```
3. Save and exit. The key rotation is disabled for the PVC.

Enabling encryption using Thales CipherTrust Manager (manual part)

Configure the Key Management Interoperability Protocol (KMIP) settings in Thales CipherTrust Manager server.

About this task

IBM Fusion HCI with IBM Spectrum® Storage Scale Erasure Code Edition (ECE) does not support the Thales or Vormetric CipherTrust Manager, and that the current encryption CR implementation in CNSA or IBM Fusion HCI works only with IBM Security GKLM (with simplified setup).

Procedure

1. If KMIP client does not exist, then create it.
   a. From the Thales CipherTrust Manager user interface, select Products > KMIP > Client Profile > Add Profile.
   b. Add the username location to the Common Name (CN) field during profile creation.
2. Create a token.
   a. Go to KMIP > Registration Token > New Registration Token.
   b. Copy the token for the next step.
3. Register the client.
   a. Go to KMIP > Registered Clients > Add Client.
   b. Specify the name.
   c. Paste the Registration Token from the previous step.
   d. Click Save.
4. To download the Private Key and Client Certificate, click Save Private Key and Save Certificate respectively.
5. Create a KMIP interface.
   a. Go to Admin Settings > Interfaces > Add Interface.
   b. Select KMIP Key Management Interoperability Protocol and click Next.
   c. Select an available Port.
   d. Select Network Interface as `all`.
   e. Select `Interface Mode` for TLS. Verify client certificate and the username that is taken from the client certificate. The auth request is optional.
   f. Select the CA to be used, and click Save.
6. To get the server CA certificate, click ellipsis overflow menu of the newly created interface, and click Download Certificate.
   Whenever you configure encryption from the IBM Fusion user interface, use these downloaded files.

Deploying Data Foundation in external mode

Fusion Data Foundation can make services from an external IBM Storage Ceph cluster available for consumption through OpenShift Container Platform clusters.

Before you begin

- Ensure to have access to an OpenShift Container Platform cluster version 4.16 or higher using an account with **cluster-admin** and operator installation permissions.
- For additional resource requirements, see Planning your deployment.
  Important:
  When you need to override the cluster-wide default node selector for Fusion Data Foundation, you can use the following command to specify a blank node selector for the **openshift-storage** namespace (create **openshift-storage** namespace in this case):

  ```
  oc annotate namespace openshift-storage openshift.io/node-selector=
  ```

- IBM Storage Ceph must have Ceph Dashboard installed and configured. For more information, see **Dashboard** > **Ceph Dashboard installation and access** within IBM Storage Ceph documentation.
- It is recommended that the external IBM Storage Ceph cluster has the PG Autoscaler enabled.

Procedure

1. Install the IBM Fusion Data Foundation operator.
    a. From the Red Hat OpenShift Container Platform web management console, go to Operators > Operator Hub and search for `IBM Fusion Data Foundation`.
    b. Click IBM Fusion Data Foundation and then click Install.
    c. Ensure that the Enable option is selected for the Console plugin.
    d. Retain all the other default settings:
        - Select the Update Channel.
        - Set Installation Mode to A specific namespace on the cluster.
        - Set Installed Namespace as Operator recommended namespace, **`openshift-storage`**. If Namespace **`openshift-storage`** does not exist, it is created during the operator installation.
        - Select Approval Strategy as Automatic or Manual.

          Automatic
          : If you select Automatic updates, then the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without any intervention.

          Manual
          : If you select Manual updates, then the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to update the Operator to a newer version.

    e. Click Install.
2. Verify the operator installation.
    a. After the operator is successfully installed, a pop-up with a message, `Web console update is available` appears on the user interface. Click Refresh web console from this pop-up for the console changes to reflect.
    b. From the Web Console do one of the following:
        - Go to Installed Operators and verify that the IBM Fusion Data Foundation Operator shows a green tick indicating successful installation.
        - Go to Storage and verify if Data Foundation dashboard is available.
3. Create a StorageSystem.
    a. View all installed operators.
    Operators > Installed Operators. Ensure that the Project selected is **`openshift-storage`**.
    b. Click the installed operator IBM Fusion Data Foundation and then click Create StorageSystem.
    c. In the Backing storage page, select the following options:
        - Select Full deployment for the Deployment type option.
        - Select Connect an external storage platform from the available options.
        - Select IBM Storage Ceph for Storage platform.
    d. Click Next.
    e. In the Connection details page, provide the necessary information:
        i. Click the Download Script link to download the python script for extracting Ceph cluster details.
        ii. For extracting the IBM Storage Ceph cluster details, contact the IBM Storage Ceph administrator to run the downloaded python script on a IBM Storage Ceph node with the **`admin key`**.
            1. Run the following command on the IBM Storage Ceph node to view the list of available arguments:

               ```
               python3 ceph-external-cluster-details-exporter.py --help
               ```

               Important: Use **`python`** instead of **`python3`** if the Ceph Storage cluster is deployed on Red Hat Enterprise Linux 7.x (RHEL 7.x) cluster.
               You can also run the script from inside a MON container (containerized deployment) or from a MON node (RPM deployment).

               Note: Use the **yum install cephadm** command and then the **cephadm** command to deploy your IBM Storage Ceph cluster using containers. You must pull the IBM Storage Ceph cluster container images using the **`cephadm`** command, rather than using **yum** for installing the Ceph packages onto nodes.
               For more information, see IBM Storage Ceph documentation.

            2. To retrieve the external cluster details from the IBM Storage Ceph cluster, use one of the following options:
                - Use the config-file flag. This stores the parameters used during deployment.

                  In new deployments, you can save the parameters used during deployment in a configuration file. This file can be used during the upgrade to preserve the parameters as well as add any additional parameters. Use the config-file flag to set the path to the configuration file.

                  An example of a configuration saved in the */config.ini* is as follows:

                  ```
                  [Configurations]
                  format = bash
                  cephfs-filesystem-name = <filesystem-name>
                  rbd-data-pool-name = <pool_name>
                  ...
                  ```

                  Run the following command to set the path to the */config.ini* using the config-file flag:

                  ```
                  python3 ceph-external-cluster-details-exporter.py --config-file /config.ini
                  ```

                - Retrieve the external cluster details from the IBM Storage Ceph cluster and pass the required parameters mentioned in Table 1 for your deployment.

                  ```
                  python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name <rbd block pool name>
                  [optional arguments]
                  ```

                  For example:

```
python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name ceph-rbd --monitoring-endpoint xxx.xxx.xxx.xxx --monitoring-endpoint-port xxxx --rgw-endpoint xxx.xxx.xxx.xxx:xxxx --run-as-user client.ocs
```

Table 1. List of parameters

Parameter type	Parameter name	Description
RBD parameters	rbd-data-pool-name	A mandatory parameter that is used for providing block storage in Fusion Data Foundation.
	rados-namespace	Divides an RBD data pool into separate logical namespaces, used for creating RBD PVC in a **radosNamespace**. Flags required with **rados-namespace** are **restricted-auth-permission** and **k8s-cluster-name**.
	rbd-metadata-ec-pool-name	(Optional) The name of the erasure coded RBD metadata pool.
RGW parameters	rgw-endpoint	(Optional) This parameter is required only if the object storage is to be provisioned through Ceph Rados Gateway for Fusion Data Foundation. Provide the endpoint in the following format: **<ip_address>:<port>** Note: A fully-qualified domain name (FQDN) is also supported in the format **<FQDN>:<PORT>**.
	rgw-pool-prefix	(Optional) The prefix of the RGW pools. If not specified, the default prefix is **default**.
	rgw-tls-cert-path	(Optional) The file path of the RADOS Gateway endpoint TLS certificate. To provide the TLS certificate and RGW endpoint details to the helper script, **ceph-external-cluster-details-exporter.py**, run the following command: **# python3 ceph-external-clustergw-endpoint r-details-exporter.py --rbd-data-pool-name <rbd block pool name> --rgw-endpoint <ip_address>:<port> --rgw-tls-cert-path <file path containing cert>** This creates a resource to create a Ceph Object Store CR such as Kubernetes secret containing the TLS certificate. All the intermediate certificates including private keys need to be stored in the certificate file
	rgw-skip-tls	(Optional) This parameter ignores the TLS certification validation when a self-signed certificate is provided (not recommended).
Monitoring parameters	monitoring-endpoint	(Optional) This parameter accepts comma-separated list of IP addresses of active and standby **mgrs** reachable from the OpenShift Container Platform cluster. If not provided, the value is automatically populated.
	monitoring-endpoint-port	(Optional) It is the port associated with the **ceph-mgr** Prometheus exporter specified by **--monitoring-endpoint**. If not provided, the value is automatically populated.
Ceph parameters	ceph-conf	(Optional) The name of the Ceph configuration file.
	run-as-user	(Optional) This parameter is used for providing name for the Ceph user which is created by the script. If this parameter is not specified, a default user name **client.healthchecker** is created. The permissions for the new user is set as: • caps: [mgr] allow command config • caps: [mon] allow r, allow command quorum_status, allow command version • caps: [osd] allow rwx pool=**RGW_POOL_PREFIX.rgw.meta**, allow r pool=**.rgw.root**, allow rw pool=**RGW_POOL_PREFIX.rgw.control**, allow rx pool=**RGW_POOL_PREFIX.rgw.log**, allow x pool=**RGW_POOL_PREFIX.rgw.buckets.index**
CephFS parameters	cephfs-metadata-pool-name	(Optional) The name of the CephFS metadata pool.
	cephfs-data-pool-name	(Optional) The name of the CephFS data pool.
	cephfs-filesystem-name	(Optional) The name of the CephFS filesystem.
Output parameters	dry-run	(Optional) This parameter helps to print the executed commands without running them.
	output	(Optional) The file where the output is required to be stored.
Multicluster parameters	k8s-cluster-name	(Optional) Kubernetes cluster name.
	cluster-name	(Optional) The Ceph cluster name.
	restricted-auth-permission	(Optional) This parameter restricts **cephCSIKeyrings** auth permissions to specific pools and clusters. Mandatory flags that need to be set with this are **rbd-data-pool-name** and **cluster-name**. You can also pass the **cephfs-filesystem-name** flag if there is CephFS user restriction so that permission is restricted to a particular CephFS filesystem. Note: This parameter must be applied only for the new deployments. To restrict **csi-users**per pool and per cluster, you need to create new **csi-users** and new secrets for those **csi-users**.

Example with restricted auth permission:

```
python3 /etc/ceph/create-external-cluster-resources.py --cephfs-filesystem-name myfs --rbd-data-pool-name replicapool --cluster-name rookStorage --restricted-auth-permission true
```

Example of JSON output generated using the python script:

```
[{"name": "rook-ceph-mon-endpoints", "kind": "ConfigMap", "data": {"data": "xxx.xxx.xxx.xxx:xxxx", "maxMonId": "0", "mapping": "{}"}}, {"name": "rook-ceph-mon", "kind": "Secret", "data": {"admin-secret": "admin-secret", "fsid": "<fs-id>", "mon-secret": "mon-secret"}}, {"name": "rook-ceph-operator-creds", "kind": "Secret", "data": {"userID": "<user-id>", "userKey": "<user-key>"}}, {"name": "rook-csi-rbd-node", "kind": "Secret", "data": {"userID": "csi-rbd-node", "userKey": "<user-key>"}}, {"name": "ceph-rbd", "kind": "StorageClass", "data": {"pool": "<pool>"}}, {"name": "monitoring-
```

```
endpoint", "kind": "CephCluster", "data": {"MonitoringEndpoint": "xxx.xxx.xxx.xxx", "MonitoringPort":
"xxxx"}}, {"name": "rook-ceph-dashboard-link", "kind": "Secret", "data": {"userID": "ceph-dashboard-
link", "userKey": "<user-key>"}}, {"name": "rook-csi-rbd-provisioner", "kind": "Secret", "data":
{"userID": "csi-rbd-provisioner", "userKey": "<user-key>"}}, {"name": "rook-csi-cephfs-provisioner",
"kind": "Secret", "data": {"adminID": "csi-cephfs-provisioner", "adminKey": "<admin-key>"}}, {"name":
"rook-csi-cephfs-node", "kind": "Secret", "data": {"adminID": "csi-cephfs-node", "adminKey": "<admin-
key>"}}, {"name": "cephfs", "kind": "StorageClass", "data": {"fsName": "cephfs", "pool":
"cephfs_data"}}, {"name": "ceph-rgw", "kind": "StorageClass", "data": {"endpoint":
"xxx.xxx.xxx.xxx:xxxx", "poolPrefix": "default"}}, {"name": "rgw-admin-ops-user", "kind": "Secret",
"data": {"accessKey": "<access-key>", "secretKey": "<secret-key>"}}]
```

3. To retrieve the external cluster details from the IBM Storage Ceph cluster, run the following command:

   ```
   python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name <rbd block pool name>  [optional
   arguments]
   ```

   For example:

   ```
   python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name ceph-rbd --monitoring-endpoint
   xxx.xxx.xxx.xxx --monitoring-endpoint-port xxxx --rgw-endpoint xxx.xxx.xxx.xxx:xxxx --run-as-user
   client.ocs
   ```

   Example with restricted auth permission:

   ```
   python3 /etc/ceph/create-external-cluster-resources.py --cephfs-filesystem-name myfs --rbd-data-pool-name
   replicapool --cluster-name rookStorage --restricted-auth-permission true
   ```

   Example of JSON output generated using the python script:

   ```
   [{"name": "rook-ceph-mon-endpoints", "kind": "ConfigMap", "data": {"data": "xxx.xxx.xxx.xxx:xxxx",
   "maxMonId": "0", "mapping": "{}"}}, {"name": "rook-ceph-mon", "kind": "Secret", "data": {"admin-secret":
   "admin-secret", "fsid": "<fs-id>", "mon-secret": "mon-secret"}}, {"name": "rook-ceph-operator-creds",
   "kind": "Secret", "data": {"userID": "<user-id>", "userKey": "<user-key>"}}, {"name": "rook-csi-rbd-node",
   "kind": "Secret", "data": {"userID": "csi-rbd-node", "userKey": "<user-key>"}}, {"name": "ceph-rbd",
   "kind": "StorageClass", "data": {"pool": "<pool>"}}, {"name": "monitoring-endpoint", "kind": "CephCluster",
   "data": {"MonitoringEndpoint": "xxx.xxx.xxx.xxx", "MonitoringPort": "xxxx"}}, {"name": "rook-ceph-
   dashboard-link", "kind": "Secret", "data": {"userID": "ceph-dashboard-link", "userKey": "<user-key>"}},
   {"name": "rook-csi-rbd-provisioner", "kind": "Secret", "data": {"userID": "csi-rbd-provisioner", "userKey":
   "<user-key>"}}, {"name": "rook-csi-cephfs-provisioner", "kind": "Secret", "data": {"adminID": "csi-cephfs-
   provisioner", "adminKey": "<admin-key>"}}, {"name": "rook-csi-cephfs-node", "kind": "Secret", "data":
   {"adminID": "csi-cephfs-node", "adminKey": "<admin-key>"}}, {"name": "cephfs", "kind": "StorageClass",
   "data": {"fsName": "cephfs", "pool": "cephfs_data"}}, {"name": "ceph-rgw", "kind": "StorageClass", "data":
   {"endpoint": "xxx.xxx.xxx.xxx:xxxx", "poolPrefix": "default"}}, {"name": "rgw-admin-ops-user", "kind":
   "Secret", "data": {"accessKey": "<access-key>", "secretKey": "<secret-key>"}}]
   ```

4. Save the JSON output to a file with `.json` extension.
   Note: For Fusion Data Foundation to work seamlessly, ensure that the parameters (RGW endpoint, CephFS details, RBD pool, and so on) to be uploaded using the JSON file remains unchanged on the IBM Storage Ceph external cluster after the storage cluster creation.
5. Run the command when there is a multi-tenant deployment in which IBM Storage Ceph cluster is already connected to Fusion Data Foundation deployment with a lower version.

   ```
   python3 ceph-external-cluster-details-exporter.py --upgrade
   ```

iii. Click Browse to select and upload the JSON file.

   The content of the JSON file is populated and displayed in the text box.

iv. Click Next
   The Next button is enabled only after you upload the JSON file.

f. Review if all the details are correct from the Review and create page.
   To modify any configuration settings, click Back to go back to the previous configuration page.
g. Click Create StorageSystem.

4. Verify the StorageSystem creation
   a. From the OpenShift Web Console, navigate to Installed Operators > IBM Fusion Data Foundation > Storage System > ocs-external-storagecluster-storagesystem > Resources.
   b. Verify that `StorageCluster` is in a *Ready* state and has a green tick.

What to do next

Deploy IBM Storage Fusion. Follow the deployment steps as documented in Deploying IBM Storage Fusion.

Overview of multiple storage cluster deployments

Multiple storage cluster deployments support one internal cluster in the `openshift-storage` namespace and one external cluster in the `openshift-storage-extended` namespace across supported platforms.

IBM Fusion Data Foundation provides the ability to deploy two storage clusters, where one is internal mode and the other is in external mode. This can be achieved only with the first cluster installed as internal in `openshift-storage` namespace while the second cluster is installed as the external in `openshift-storage-extended` namespace. Clusters installed conversely is not currently supported.

Supported platforms

- Bare metal
- VMware VSphere
- OpenStack

- OpenShift Virtualization
- IBM Cloud
- IBM Power

Preparing to deploy multiple Fusion Data Foundation storage clusters

Prerequisites, namespace restrictions, and configuration constraints to review before deploying multiple Fusion Data Foundation storage clusters.

Before you begin the deployment of Fusion Data Foundation using dynamic, local, or external storage, ensure that your resource requirements are met. See the Resource requirements section in the Planning guide.

Things you should remember before installing multiple Fusion Data Foundation storage clusters:

- `openshift-storage` and `openshift-storage-extended` are the exclusively supported namespaces.
- Internal storage cluster is restricted to the Fusion Data Foundation operator namespace.
- External storage cluster is permissible in both operator and non-operator namespaces.
- Multiple storage clusters are not supported in the same namespace. Hence, the external storage system is not visible under the Fusion Data Foundation operator page as the operator is under `openshift-storage` namespace and the external storage system is not.
- Customers running external storage clusters in the operator namespace cannot utilize multiple storage clusters.
- **Multicloud Object Gateway** is supported solely within the operator namespace. It is ignored in other namespaces.
- **RADOS Gateway (RGW)** can be in either the operator namespace, a non-operator namespace, or both
- **Network File System (NFS)** is enabled as long as it is enabled for at least one of the clusters.
- Topology is enabled as long as it is enabled for at least one of the clusters.
- Topology domain labels are set as long as the internal cluster is present.
- The **Topology** view of the cluster is only supported for Fusion Data Foundation internal mode deployments.
- Different multus settings are not supported for multiple storage clusters.

To deploy and verify Fusion Data Foundation storage cluster in the internal mode, refer to your respective infrastructure deployment guides.

To deploy Fusion Data Foundation in internal or external mode, see the following topics:

- Deploying IBM Fusion
- Deploying Data Foundation external storage cluster

Deploying Data Foundation external storage cluster

Use this procedure to deploy an external storage cluster to add additional storage or expand your current internal storage cluster.

Before you begin

- An Fusion Data Foundation cluster deployed in internal mode.
- Ensure that both the OpenShift container Platform and Fusion Data Foundation are upgraded to version 4.16 or later.

Procedure

1. Log in to the Red Hat® OpenShift® Container Platform web console as an administrator, and navigate to Storage > External systems.
2. Click Connect external systems.
3. Select Red Hat/IBM Ceph Cluster and click Next.
4. In the Connect Red Hat Ceph storage page, do as follows:
    a. Click the Download script link to download the Python script for extracting Ceph cluster details.
    b. To extract the IBM Storage Ceph cluster details, contact the IBM Storage Ceph administrator to run the downloaded Python script on an IBM Storage Ceph node by using the `admin key`.
        i. Run the following command on the IBM Storage Ceph node to view the list of available arguments:

        ```
        python3 ceph-external-cluster-details-exporter.py --help
        ```

        Important: Use `python` instead of `python3` if the IBM Storage Ceph cluster is deployed on a Red Hat Enterprise Linux 7.x (RHEL 7.x) cluster. You can also run the script from inside a MON container (for containerized deployments) or from a MON node (for RPM deployments).

        Note: Use the **yum install cephadm** command, followed by the **cephadm** command to deploy your IBM Storage Ceph cluster using containers. You must pull the IBM Storage Ceph cluster container images by using the `cephadm` command, rather than installing Ceph packages on nodes by using the **yum** command. For more information, see IBM Storage Ceph documentation.
        ii. Retrieve the external cluster details from the IBM Storage Ceph cluster by using one of the following options:

        Configuration file flag
        Use the config-file flag that stores the parameters used during deployment.

        In new deployments, you can save the deployment parameters in a configuration file. This file can also be used during upgrades to preserve existing parameters and add any new parameters. Use the config-file flag to specify the path to the configuration file.

        An example of a configuration saved in the /config.ini file is as follows:

        ```
        [Configurations]
        format = bash
        cephfs-filesystem-name = <filesystem-name>
        ```

```
rbd-data-pool-name = <pool_name>
...
```

Run the following command to set the path to the /config.ini by using the config-file flag:

```
python3 ceph-external-cluster-details-exporter.py --config-file /config.ini
```

Command line flag

Retrieve the external cluster details from the IBM Storage Ceph cluster and configure the required parameters mentioned in Table 1 for your deployment.

Command example:

```
python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name ceph-rbd --monitoring-endpoint xxx.xxx.xxx.xxx --monitoring-endpoint-port xxxx --rgw-endpoint xxx.xxx.xxx.xxx:xxxx --run-as-user client.ocs
```

Table 1. List of parameters

Parameter type	Parameter name	Description
RBD parameters	rbd-data-pool-name	A mandatory parameter that is used for providing block storage in Fusion Data Foundation.
	rados-namespace	Divides an RBD data pool into separate logical namespaces, used for creating RBD PVC in a **`radosNamespace`**. Flags required with **`rados-namespace`** are **`restricted-auth-permission`** and **`k8s-cluster-name`**.
	rbd-metadata-ec-pool-name	(Optional) The name of the erasure coded RBD metadata pool.
RGW parameters	rgw-endpoint	(Optional) This parameter is required only if the object storage is to be provisioned through Ceph Rados Gateway for Fusion Data Foundation. Provide the endpoint in the following format: **`<ip_address>:<port>`** Note: A fully-qualified domain name (FQDN) is also supported in the format **`<FQDN>:<PORT>`**.
	rgw-pool-prefix	(Optional) The prefix of the RGW pools. If not specified, the default prefix is **`default`**.
	rgw-tls-cert-path	(Optional) The file path of the RADOS Gateway endpoint TLS certificate. To provide the TLS certificate and RGW endpoint details to the helper script, **`ceph-external-cluster-details-exporter.py`**, run the following command: **`# python3 ceph-external-clustergw-endpoint r-details-exporter.py --rbd-data-pool-name <rbd block pool name> --rgw-endpoint <ip_address>:<port> --rgw-tls-cert-path <file path containing cert>`** This creates a resource to create a Ceph Object Store CR such as Kubernetes secret containing the TLS certificate. All the intermediate certificates including private keys need to be stored in the certificate file
	rgw-skip-tls	(Optional) This parameter ignores the TLS certification validation when a self-signed certificate is provided (not recommended).
Monitoring parameters	monitoring-endpoint	(Optional) This parameter accepts comma-separated list of IP addresses of active and standby **`mgrs`** reachable from the OpenShift Container Platform cluster. If not provided, the value is automatically populated.
	monitoring-endpoint-port	(Optional) It is the port associated with the **`ceph-mgr`** Prometheus exporter specified by **`--monitoring-endpoint`**. If not provided, the value is automatically populated.
Ceph parameters	ceph-conf	(Optional) The name of the Ceph configuration file.
	run-as-user	(Optional) This parameter is used for providing name for the Ceph user which is created by the script. If this parameter is not specified, a default user name **`client.healthchecker`** is created. The permissions for the new user is set as: • caps: [mgr] allow command config • caps: [mon] allow r, allow command quorum_status, allow command version • caps: [osd] allow rwx pool=**`RGW_POOL_PREFIX.rgw.meta`**, allow r pool=**`.rgw.root`**, allow rw pool=**`RGW_POOL_PREFIX.rgw.control`**, allow rx pool=**`RGW_POOL_PREFIX.rgw.log`**, allow x pool=**`RGW_POOL_PREFIX.rgw.buckets.index`**
CephFS parameters	cephfs-metadata-pool-name	(Optional) The name of the CephFS metadata pool.
	cephfs-data-pool-name	(Optional) The name of the CephFS data pool.
	cephfs-filesystem-name	(Optional) The name of the CephFS filesystem.
Output parameters	dry-run	(Optional) This parameter helps to print the executed commands without running them.
	output	(Optional) The file where the output is required to be stored.
Multicluster parameters	k8s-cluster-name	(Optional) Kubernetes cluster name.
	cluster-name	(Optional) The Ceph cluster name.
	restricted-auth-permission	(Optional) This parameter restricts **`cephCSIKeyrings`** auth permissions to specific pools and clusters. Mandatory flags that need to be set with this are **`rbd-data-pool-name`** and **`cluster-name`**. You can also pass the **`cephfs-filesystem-name`** flag if there is CephFS user restriction so that permission is restricted to a particular CephFS filesystem. Note: This parameter must be applied only for the new deployments. To restrict **`csi-users`**per pool and per cluster, you need to create new **`csi-users`** and new secrets for those **`csi-users`**.

Command example with restricted auth permission:

```
python3 /etc/ceph/create-external-cluster-resources.py --cephfs-filesystem-name myfs --rbd-data-pool-name
replicapool --cluster-name rookStorage --restricted-auth-permission true
```

Command example of JSON output generated by using the Python script:

```
[{"name": "rook-ceph-mon-endpoints", "kind": "ConfigMap", "data": {"data": "xxx.xxx.xxx.xxx:xxxx",
"maxMonId": "0", "mapping": "{}"}}, {"name": "rook-ceph-mon", "kind": "Secret", "data": {"admin-secret":
"admin-secret", "fsid": "<fs-id>", "mon-secret": "mon-secret"}}, {"name": "rook-ceph-operator-creds",
"kind": "Secret", "data": {"userID": "<user-id>", "userKey": "<user-key>"}}, {"name": "rook-csi-rbd-node",
"kind": "Secret", "data": {"userID": "csi-rbd-node", "userKey": "<user-key>"}}, {"name": "ceph-rbd",
"kind": "StorageClass", "data": {"pool": "<pool>"}}, {"name": "monitoring-endpoint", "kind": "CephCluster",
"data": {"MonitoringEndpoint": "xxx.xxx.xxx.xxx", "MonitoringPort": "xxxx"}}, {"name": "rook-ceph-
dashboard-link", "kind": "Secret", "data": {"userID": "ceph-dashboard-link", "userKey": "<user-key>"}},
{"name": "rook-csi-rbd-provisioner", "kind": "Secret", "data": {"userID": "csi-rbd-provisioner", "userKey":
"<user-key>"}}, {"name": "rook-csi-cephfs-provisioner", "kind": "Secret", "data": {"adminID": "csi-cephfs-
provisioner", "adminKey": "<admin-key>"}}, {"name": "rook-csi-cephfs-node", "kind": "Secret", "data":
{"adminID": "csi-cephfs-node", "adminKey": "<admin-key>"}}, {"name": "cephfs", "kind": "StorageClass",
"data": {"fsName": "cephfs", "pool": "cephfs_data"}}, {"name": "ceph-rgw", "kind": "StorageClass", "data":
{"endpoint": "xxx.xxx.xxx.xxx:xxxx", "poolPrefix": "default"}}, {"name": "rgw-admin-ops-user", "kind":
"Secret", "data": {"accessKey": "<access-key>", "secretKey": "<secret-key>"}}]
```

iii. Save the JSON output to a file with .json extension.
Note: For Fusion Data Foundation to work seamlessly, ensure that the parameters (RGW endpoint, CephFS details, RBD pool, and so on) uploaded by using the JSON file remain unchanged on the external IBM Storage Ceph cluster after the storage cluster is created.

iv. Run the command in a multi-tenant deployment where the IBM Storage Ceph cluster is already connected to an IBM Storage Ceph deployment with an earlier version.

```
python3 ceph-external-cluster-details-exporter.py --upgrade
```

c. Click Browse to select and upload the JSON file.
The JSON file content is populated and displayed in the text box.

5. Optional: Select In-transit encryption to enable encryption for entire Ceph traffic including data using Ceph `msgrv2`.
Note: Verify that the external IBM Storage Ceph cluster is configured for in-transit encryption. Before enabling this setting, use the documentation link in the UI to complete the required configuration steps.
   a.
      i.
         1. (Optional) Select the advanced configuration options that apply to your environment:

      ii. Review the selections and click **Connect**.
6. Optional: Select the required Advanced configuration option that apply to your environment:
   - Use Ceph RBD as the default StorageClass

     Sets the RBD StorageClass as the cluster default.

   - Use external PostgreSQL *(Technology preview)*

     Uses an external PostgreSQL instance instead of the in-cluster database.

     Note: Additional advanced options may be available in the UI. Select only the options that are required for the deployment policy and environment.
   - Enable Automatic backup

     Allows automatic backup for the Multicloud Object Gateway metadata database.

     - Select the backup frequency: Daily, Weekly, or Monthly.
     - Specify the number of backups to retain.

       Note: This option is disabled if you select Use external PostgreSQL.
7. Review the details, and then click Connect.

What to do next

Verify the StorageSystem creation.

1. Go to Storage > Data Foundation > Storage System tab and verify that you can view all storage clusters.
2. Verify that all components for the external Fusion Data Foundation are successfully installed. See Verifying external Data Foundation storage cluster deployment for instructions.

Verifying external Data Foundation storage cluster deployment

Use this section to verify that the Fusion Data Foundation deployed as external storage is deployed correctly.

- **Verifying the state of the pods**
  Use this procedure to verify the state of the pods.
- **Verifying the Fusion Data Foundation cluster is healthy**
  Verify that the Fusion Data Foundation cluster is healthy by checking the Storage Cluster status and cluster details in the OpenShift Web Console.
- **Verifying the Multicloud Object Gateway is healthy**
  Follow this procedure to verify that the Multicloud Object Gateway is healthy.
- **Verifying that the specific storage classes exist**
  Use this procedure to verify that storage classes are created with the Fusion Data Foundation cluster creation.
- **Verifying that Ceph cluster is connected**
  Verify that the Fusion Data Foundation cluster is connected to the external Red Hat Ceph Storage cluster and shows a `Connected` status with `HEALTH_OK`.

- **Verifying that storage cluster is ready**
  Verify that the external storage cluster is in a `Ready` state and that the `External` option is set to `true`.

Verifying the state of the pods

Use this procedure to verify the state of the pods.

Procedure

1. Click Workloads → Pods from the OpenShift Web Console.
2. Select openshift-storage from the Project drop-down list.
   Note: If the Show default projects option is disabled, use the toggle button to list all the default projects.
   For more information on the expected number of pods for each component and how it varies depending on the number of nodes, see Table 1.
3. Click the Running and Completed tabs to verify that the following pods are in `Running` and `Completed` state:

Table 1. Pods corresponding to Fusion Data Foundation cluster components

Component	Corresponding pods
Fusion Data Foundation Operator	• `ocs-operator-*` (1 pod on any worker node) • `ocs-metrics-exporter-*` (1 pod on any worker node) Note: This pod must be present in `openshift-storage-extended` namespace as well such that there is 1 pod in each `openshift-storage` and `openshift-storage extended` namespace. • `odf-operator-controller-manager-*` (1 pod on any worker node) • `odf-console-*` (1 pod on any worker node) • `csi-addons-controller-manager-*` (1 pod on any worker node)
Rook-ceph Operator	`rook-ceph-operator-*` (1 pod on any worker node)

CSI	• `cephfs`
	  • `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*` (1 pod on each worker node)
	  • `openshift-storage.cephfs.csi.ceph.com-nodeplugin-*` (2 pods distributed across worker nodes)
	• `rbd`
	  • `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*` (1 pod on each worker node)
	  • `openshift-storage.rbd.csi.ceph.com-nodeplugin-*` (2 pods distributed across worker nodes)

ceph-csi-operator	`ceph-csi-controller-manager-*` (1 pod for each device)

Verifying the Fusion Data Foundation cluster is healthy

Verify that the Fusion Data Foundation cluster is healthy by checking the Storage Cluster status and cluster details in the OpenShift Web Console.

Procedure

1. In the OpenShift Web Console, click Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the notification window that appears.
3. In the Status card of the Block and File tab, verify that *Storage Cluster* has a green tick.
4. In the Details card, verify that the cluster information is displayed.

What to do next

For more information about the health of the Fusion Data Foundation cluster on the Block and File dashboard, see Monitoring Fusion Data Foundation.

Verifying the Multicloud Object Gateway is healthy

Follow this procedure to verify that the Multicloud Object Gateway is healthy.

Procedure

1. In the OpenShift Web Console, click Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the notification window that appears.
   a. In the Status card of the Object tab, verify that both *Object Service* and *Data Resiliency* have a green tick.
   b. In the Details card, verify that the MCG information is displayed.

   Note: The RADOS Object Gateway is only listed in case RADOS Object Gateway endpoint details are included while deploying Fusion Data Foundation in external mode.

What to do next

For more information about the health of the Fusion Data Foundation cluster on the object service dashboard, see Monitoring Fusion Data Foundation.

Verifying that the specific storage classes exist

Use this procedure to verify that storage classes are created with the Fusion Data Foundation cluster creation.

Procedure

1. Click Storage > Storage Classes from the left pane of the Red Hat OpenShift Web Console.
2. Verify that the following storage classes are created with the Fusion Data Foundation cluster creation:
   - `ocs-storagecluster-ceph-rbd`
   - `ocs-storagecluster-ceph-rgw`
   - `ocs-storagecluster-cephfs`

   Note:
   - If an MDS is not deployed in the external cluster, `ocs-external-storagecluster-cephfs` storage class is not created.
   - If RGW is not deployed in the external cluster, the `ocs-external-storagecluster-ceph-rgw` storage class is not created.

   For more information regarding MDS and RGW, see IBM Storage Ceph documentation.

Verifying that Ceph cluster is connected

Verify that the Fusion Data Foundation cluster is connected to the external Red Hat Ceph Storage cluster and shows a **`Connected`** status with **`HEALTH_OK`**.

Run the following command to verify if the Fusion Data Foundation cluster is connected to the external Red Hat Ceph Storage cluster.

```
oc get cephcluster -n openshift-storage-extended
```

```
NAME                                      DATADIRHOSTPATH   MONCOUNT   AGE   PHASE       MESSAGE
HEALTH      EXTERNAL   FSID
ocs-external-storagecluster-cephcluster                                51m   Connected   Cluster connected successfully
HEALTH_OK   true       8f01d842-d4b2-11ee-b43c-0050568fb522
```

Verifying that storage cluster is ready

Verify that the external storage cluster is in a **`Ready`** state and that the **`External`** option is set to **`true`**.

Run the following command to verify if the storage cluster is ready and the **`External`** option is set to **`true`**.

```
oc get storagecluster -n openshift-storage-extended
```

```
NAME                          AGE   PHASE   EXTERNAL   CREATED AT             VERSION
ocs-external-storagecluster   51m   Ready   true       2024-02-28T10:05:54Z   4.15.0
```

Migrating application workloads

Migrate application workloads from internal mode storage classes to external mode storage classes using the Migration Toolkit for Containers on the same cluster.

You can migrate application workloads from the internal mode storage classes to the external mode storage classes using the same cluster as source and target. For more information, see Update and migrate > Migration Toolkit for Containers in Red Hat OpenShift Container Platform product documentation.

Managing and allocating resources

Understand how to create, configure, and allocate storage to core services or hosted applications in IBM Fusion Data Foundation.

- **Storage classes**
  Use this information to understand how to create custom storage classes.
- **Persistent volume encryption**
  Persistent volume (PV) encryption protects tenant data by securing RADOS Block Device (RBD) PVs through an encryption-enabled storage class. Configure access to an external key management system (KMS), such as Vault or CipherTrust, before creating the storage class.
- **Enabling and disabling encryption in-transit post deployment**
  You can enable encryption in-transit for the existing clusters after the deployment of clusters both in internal and external modes.
- **Block pools**
  This section provides you with information about how to configure block pools in internal mode.
- **Configure storage for OpenShift Container Platform services**
  Use Fusion Data Foundation for core OpenShift Container Platform services, such as image registry, monitoring, and logging.
- **Creating Multus networks**
  Multus networks enable pods to attach additional network interfaces by creating **`NetworkAttachmentDefinition`** custom resources; Fusion Data Foundation

uses the macvlan CNI plug-in to provide each pod with a unique MAC address on the additional network.

- **Backing OpenShift Container Platform applications with Fusion Data Foundation**
  Configure OpenShift Container Platform applications to use Fusion Data Foundation.
- **Adding file and object storage to an existing external Fusion Data Foundation cluster**
  Add file storage (using Metadata Servers) or object storage (using Ceph Object Gateway) or both to an external Fusion Data Foundation cluster that was initially deployed to provide only block storage.
- **How to use dedicated worker nodes for Fusion Data Foundation**
  Use dedicated worker nodes for IBM Fusion Data Foundation
- **Managing persistent volume claims**
  Manage and automate the fulfillment of Persistent Volume Claim requests.
- **Reclaiming space on target volumes**
  Reclaim the actual available storage space on target volumes.
- **Finding and cleaning stale subvolumes**
  This section describes how to find and clean stale subvolumes.
- **Volume snapshots**
  Create, restore, and delete volume snapshots.
- **Volume cloning**
  A clone is a duplicate of an existing storage volume that is used as any standard volume. You create a clone of a volume to make a point in time copy of the data. A persistent volume claim (PVC) cannot be cloned with a different size. You can create up to 512 clones per PVC for both CephFS and RADOS Block Device (RBD).
- **Managing Container Storage Interface (CSI) component placements**
  You can set tolerations to run the Container Storage Interface (CSI) components on the `infra` nodes with custom taints.
- **2-way replication with CephFS**
  To reduce storage load with the Ceph File System (CephFS) when data resiliency is not a primary concern, you can choose to use 2-way replication (replica-2). It reduces the amount of storage space that is used and decreases the level of fault tolerance.
- **Creating exports using NFS**
  Create exports using NFS that can then be accessed externally from the Fusion Data Foundation cluster.
- **Annotating encrypted RBD storage classes**
  This section provides you with information on annotating encrypted RBD storage classes.
- **Enabling faster client IO or recovery IO during OSD backfill**
  Use this procedure for faster recovery during OSD backfill.
- **Setting Ceph OSD full thresholds**
  You can set Ceph Object Storage Daemon (OSD) full thresholds by using the OpenShift Data Foundation command-line interface (CLI) tool or by updating the StorageCluster Custom Resource (CR).
- **Configuring Ceph target size ratios**
  Learn how to configure `targetSizeRatio` values to optimize Ceph storage distribution across block, filesystem, and object pools according to workload requirements.
- **Configuring Ceph monitor failover timeout**
  Configure the Ceph monitor failover timeout to control when the system replaces an unhealthy monitor, providing flexibility during maintenance operations and improving cluster stability.
- **Backup and recovery of Multicloud Object Gateway metadata database (NooBaa DB)**
  Configure automated and on-demand backups for the Multicloud Object Gateway metadata database to maintain data resilience and improve recovery options.

Storage classes

Use this information to understand how to create custom storage classes.

The Fusion Data Foundation operator installs a default storage class depending on the platform in use. This default storage class is owned and controlled by the operator and it cannot be deleted or modified. However, you can create custom storage classes to use other storage resources or to offer a different behavior to applications.

Note: Custom storage classes are not supported for *external mode* Fusion Data Foundation clusters.

- **Creating storage classes and pools**
  You can create a storage class using an existing pool or you can create a new pool for the storage class while creating it.
- **Storage class with single replica**
  You can create a storage class with a single replica to be used by your applications. This avoids redundant data copies and allows resiliency management on the application level.

Creating storage classes and pools

You can create a storage class using an existing pool or you can create a new pool for the storage class while creating it.

Before you begin

Ensure that you are logged into the OpenShift Container Platform web console and Fusion Data Foundation cluster is in *Ready* state.

Procedure

1. Create a storage class by going to Storage > StorageClasses and clicking Create Storage Class. Fill in the form with the following information:
2. Fill in the storage class Name and Description.
   Remember: The storage class name must not exceed 40 characters and must not match any of the following predefined storage classes:
   - `ocs-storagecluster-cephfs`
   - `ocs-storagecluster-ceph-rbd`
   - `ocs-storagecluster-ceph-rgw`
   - `openshift-storage.noobaa.io`

- `ocs-storagecluster-ceph-rbd-virtualization`

3. Use the default Relcaim Policy, *Delete*.
   Important: If you change the reclaim policy to *Retain* in the storage class, the persistent volume (PV) remains in *Released* state even after deleting the persistent volume claim (PVC).
4. Set the Volume binding mode. The default is *WaitForConsumer*.
   Note: If you select *Immediate* option, then the PV gets created immediately when creating the PVC.
5. Select RBD or CephFS Provisioner, which is the plugin used for provisioning the persistent volumes.
6. Select a Storage system for your workloads.
7. Select an existing Storage Pool from the list or create a new pool.
   Note:
   The 2-way replication data protection policy is only supported for the non-default RBD pool. 2-way replication can be used by creating an additional pool.

   To create a new pool:
   a. Click Create New Pool.
   b. Enter Pool name.
   c. Select the Data Protection Policy, either 2-way-Replication or 3-way-Replication.
   d. Optional: If you need to compress the data, select Enable compression.
      Enabling compression can impact application performance and might prove ineffective when data to be written is already compressed or encrypted. Data written before enabling compression are not compressed.
   e. Click Create to create the new storage pool.
   f. Click Finish after the pool is created.
8. Optional: Select Enable Encryption.
9. Click Create Create to create the storage class

Storage class with single replica

You can create a storage class with a single replica to be used by your applications. This avoids redundant data copies and allows resiliency management on the application level.

About this task

Warning: Enabling this feature creates a single replica pool without data replication, increasing the risk of data loss, data corruption, and potential system instability if your application does not have its own replication. If any OSDs are lost, this feature requires very disruptive steps to recover. All applications can lose their data, and must be recreated in case of a failed OSD.

Procedure

1. Enable the single replica feature using the following command:

   ```
   $ oc patch storagecluster ocs-storagecluster -n openshift-storage --type json --patch '[{ "op": "replace", "path": "/spec/managedResources/cephNonResilientPools/enable", "value": true }]'
   ```

2. Verify `storagecluster` is in `Ready` state:

   ```
   $ oc get storagecluster
   ```

   Example output:

   ```
   NAME                 AGE   PHASE   EXTERNAL   CREATED AT             VERSION
   ocs-storagecluster   10m   Ready              2024-02-05T13:56:15Z   4.15.0
   ```

   New `cephblockpools` are created for each failure domain.
3. Verify `cephblockpools` are in `Ready` state:

   ```
   $ oc get cephblockpools
   ```

   Example output:

   ```
   NAME                                          PHASE
   ocs-storagecluster-cephblockpool              Ready
   ocs-storagecluster-cephblockpool-us-east-1a   Ready
   ocs-storagecluster-cephblockpool-us-east-1b   Ready
   ocs-storagecluster-cephblockpool-us-east-1c   Ready
   ```

4. Verify new storage classes have been created:

   ```
   $ oc get storageclass
   ```

   Example output:

   ```
   NAME                                          PROVISIONER                             RECLAIMPOLICY   VOLUMEBINDINGMODE
   ALLOWVOLUMEEXPANSION   AGE
   gp2 (default)                                 kubernetes.io/aws-ebs                   Delete          WaitForFirstConsumer
   true                   104m
   gp2-csi                                       ebs.csi.aws.com                         Delete          WaitForFirstConsumer
   true                   104m
   gp3-csi                                       ebs.csi.aws.com                         Delete          WaitForFirstConsumer
   true                   104m
   ocs-storagecluster-ceph-non-resilient-rbd     openshift-storage.rbd.csi.ceph.com      Delete          WaitForFirstConsumer
   true                   46m
   ```

```
ocs-storagecluster-ceph-rbd                openshift-storage.rbd.csi.ceph.com      Delete          Immediate
true                   52m
ocs-storagecluster-cephfs                  openshift-storage.cephfs.csi.ceph.com   Delete          Immediate
true                   52m
openshift-storage.noobaa.io                openshift-storage.noobaa.io/obc         Delete          Immediate
false                  50m
```

New OSD pods are created; 3 `osd-prepare` pods and 3 additional pods.

5. Verify new OSD pods are in `Running` state:

```
$ oc get pods | grep osd
```

Example output:

```
rook-ceph-osd-0-6dc76777bc-snhnm                                  2/2     Running     0               9m50s
rook-ceph-osd-1-768bdfdc4-h5n7k                                   2/2     Running     0               9m48s
rook-ceph-osd-2-69878645c4-bkdlq                                  2/2     Running     0               9m37s
rook-ceph-osd-3-64c44d7d76-zfxq9                                  2/2     Running     0               5m23s
rook-ceph-osd-4-654445b78f-nsgjb                                  2/2     Running     0               5m23s
rook-ceph-osd-5-5775949f57-vz6jp                                  2/2     Running     0               5m22s
rook-ceph-osd-prepare-ocs-deviceset-gp2-0-data-0x6t87-59swf       0/1     Completed   0               10m
rook-ceph-osd-prepare-ocs-deviceset-gp2-1-data-0klwr7-bk45t       0/1     Completed   0               10m
rook-ceph-osd-prepare-ocs-deviceset-gp2-2-data-0mk2cz-jx7zv       0/1     Completed   0               10m
```

Persistent volume encryption

Persistent volume (PV) encryption protects tenant data by securing RADOS Block Device (RBD) PVs through an encryption-enabled storage class. Configure access to an external key management system (KMS), such as Vault or CipherTrust, before creating the storage class.

PV encryption ensures data isolation and confidentiality between tenants (applications). Before you can use PV encryption, you must create a storage class for PV encryption. Persistent volume encryption is only available for RBD PVs.

Fusion Data Foundation supports storing encryption passphrases in HashiCorp Vault and Thales CipherTrust Manager. You can create an encryption-enabled storage class by using a KMS for persistent volume encryption. You need to configure access to the KMS before creating the storage class.

- **Access configuration for KMS**
  Key management system (KMS) access can be configured through Vault tokens, Thales CipherTrust Manager, or Vault tenants.
- **Creating a storage class for persistent volume encryption**
  Create a storage class for persistent volume (PV) encryption.
- **Enabling and disabling key rotation when using KMS**
  Security common practices require periodic encryption of key rotation. You can enable or disable key rotation when using key management system (KMS).

Access configuration for KMS

Key management system (KMS) access can be configured through Vault tokens, Thales CipherTrust Manager, or Vault tenants.

Based on your use case, you need to configure access to KMS using one of the following ways:

- Using `vaulttokens`: allows users to authenticate using a token
- Using `Thales CipherTrust Manager`: uses Key Management Interoperability Protocol (KMIP)
- Using `vaulttenantsa` (Technology Preview): allows users to use `serviceaccounts` to authenticate with `Vault`

Important: Accessing the KMS using `vaulttenantsa` is a Technology Preview feature. Technology Preview features are not supported with IBM production service level agreements (SLAs) and might not be functionally complete. IBM does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

- **Configuring access to KMS by using vaulttokens**
  Create a Vault token secret in the tenant namespace to configure access to the external key management system (KMS) and enable secure encryption operations for Fusion Data Foundation.
- **Configuring access to KMS by using Thales CipherTrust Manager**
  Configure Thales CipherTrust Manager to provide secure key management by creating the required KMIP clients and interfaces, and generating the Key Encryption Key (KEK) for storageclass encryption.
- **Configuring access by using vaulttenantsa**
  Configure the key management system (KMS) by using `vaulttenantsa` to enable authentication with Vault through a service account.

Configuring access to KMS by using vaulttokens

Create a Vault token secret in the tenant namespace to configure access to the external key management system (KMS) and enable secure encryption operations for Fusion Data Foundation.

Before you begin

- Ensure that the Fusion Data Foundation cluster is in the `Ready` state.
- On the external key management system (KMS):
  - Ensure that a policy with a token exists and that the key value backend path in `Vault` is enabled.
  - Ensure that the `Vault` servers use signed certificates.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Workloads > Secrets.
2. Click Create > Key/value secret.
3. Enter Secret Name as `ceph-csi-kms-token`.
4. Enter Key as `token`.
5. Enter Value.
   It is the token from Vault. You can either click Browse to upload the file that contains the token, or enter the token directly in the text box.
6. Click Create.
   Note: You can delete the token only after all the encrypted PVCs that use the `ceph-csi-kms-token` are deleted.

Configuring access to KMS by using Thales CipherTrust Manager

Configure Thales CipherTrust Manager to provide secure key management by creating the required KMIP clients and interfaces, and generating the Key Encryption Key (KEK) for storageclass encryption.

Before you begin

1. Create a KMIP client if one does not exist.
   a. From the user interface, select KMIP > Client Profile > Add Profile.
   b. Add the `CipherTrust` username to the Common Name field during profile creation.
2. Go to KMIP > Registration Token > New Registration Token to create a token.

   Copy the token for the next step.
3. To register the client, go to KMIP > Registered Clients > Add Client.
   a. Specify the Name.
   b. Paste the Registration Token from the previous step, and click Save.
4. Download the private key and client certificate by clicking Save Private Key and Save Certificate, respectively.
5. To create a new KMIP interface, go to Admin Settings > Interfaces > Add Interface.
   a. Select KMIP Key Management Interoperability Protocol, and click Next.
   b. Select a free Port.
   c. Select Network Interface to all.
   d. Select Interface Mode as TLS, verify client cert, user name taken from client cert, auth request is optional.
   e. (Optional) Enable hard delete to delete both metadata and material when the key is deleted.

      This option is disabled by default.
   f. Select the CA to use, and click Save.
6. To get the server CA certificate, click the Action menu (⋮) next to the newly created interface, and click Download Certificate.

Procedure

1. Go to Keys > Add Key.
2. Enter Key Name.
3. Set Algorithm to AES and Size to 256.
4. Enable Create a key in Pre-Active state, and set the date and time for activation.
5. Ensure that Encrypt and Decrypt are enabled under Key Usage.
6. Copy the ID of the newly created key to use as the unique identifier during deployment.

Configuring access by using vaulttenantsa

Configure the key management system (KMS) by using `vaulttenantsa` to enable authentication with Vault through a service account.

Before you begin

- Ensure that the Fusion Data Foundation cluster is in the `Ready` state.
- On the external key management system (KMS):
  - Ensure that a policy with a token exists and that the key value backend path in `Vault` is enabled.
  - Ensure that the `Vault` servers use signed certificates.
- Ensure that the service account is created in the tenant namespace.

  For example:

```
 cat <<EOF | oc create -f -
apiVersion: v1
kind: ServiceAccount
metadata:
    name: ceph-csi-vault-sa
EOF
```

1. The Fusion Data Foundation cluster is in *Ready* state.
2. On the external key management system (KMS):
   - Ensure that a policy exists and the key value backend path in Vault is enabled.
   - Ensure that you are using signed certificates on your Vault servers.

About this task

You must configure the Kubernetes authentication method before Fusion Data Foundation can authenticate with and start using **`Vault`**. The following instructions create and configure **`serviceAccount`**, **`ClusterRole`**, and **`ClusterRoleBinding`** required to allow Fusion Data Foundation to authenticate with **`Vault`**.

Procedure

1. Apply the following YAML to your OpenShift® cluster:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: rbd-csi-vault-token-review
---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: rbd-csi-vault-token-review
rules:
  - apiGroups: ["authentication.k8s.io"]
    resources: ["tokenreviews"]
    verbs: ["create", "get", "list"]

---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: rbd-csi-vault-token-review
subjects:
  - kind: ServiceAccount
    name: rbd-csi-vault-token-review
    namespace: openshift-storage
roleRef:
  kind: ClusterRole
  name: rbd-csi-vault-token-review
  apiGroup: rbac.authorization.k8s.io
```

2. Create a secret for `serviceaccount` token and CA certificate.

```
 cat <<EOF | oc create -f -
apiVersion: v1
kind: Secret
metadata:
  name: rbd-csi-vault-token-review-token
  namespace: openshift-storage
  annotations:
    kubernetes.io/service-account.name: "rbd-csi-vault-token-review"
type: kubernetes.io/service-account-token
data: {}
EOF
```

3. Get the token and the CA certificate from the secret.

```
$ SA_JWT_TOKEN=$(oc -n openshift-storage get secret rbd-csi-vault-token-review-token -o jsonpath="{.data['token']}" |
base64 --decode; echo)
$ SA_CA_CRT=$(oc -n openshift-storage get secret rbd-csi-vault-token-review-token -o jsonpath="{.data['ca\.crt']}" |
base64 --decode; echo)
```

4. Retrieve the OpenShift cluster endpoint.

```
OCP_HOST=$(oc config view --minify --flatten -o jsonpath="{.clusters[0].cluster.server}")
```

5. Use the information collected in the previous steps to set up the Kubernetes authentication method in Vault as shown:

```
$ vault auth enable kubernetes
$ vault write auth/kubernetes/config \
          token_reviewer_jwt="$SA_JWT_TOKEN" \
          kubernetes_host="$OCP_HOST" \
          kubernetes_ca_cert="$SA_CA_CRT"
```

6. Create a role in Vault for the tenant namespace.

```
$ vault write "auth/kubernetes/role/csi-kubernetes" bound_service_account_names="ceph-csi-vault-sa"
bound_service_account_namespaces=<tenant_namespace> policies=<policy_name_in_vault>
```

   `csi-kubernetes` is the default role name that Fusion Data Foundation looks for in Vault. The default service account name in the tenant namespace in the Fusion Data Foundation cluster is **`ceph-csi-vault-sa`**. These default values can be overridden by creating a ConfigMap in the tenant namespace.

   For more information about overriding the default names, see Overriding Vault connection details using tenant ConfigMap.

Example

To create a StorageClass that uses the **`vaulttenantsa`** method for PV encryption, you must either edit the existing ConfigMap or create a ConfigMap named **`csi-kms-connection-details`** that holds all the information needed to establish the connection with Vault.

Use this sample YAML to update or create the `csi-kms-connection-detail` ConfigMap:

```
apiVersion: v1
data:
  vault-tenant-sa: |-
    {
      "encryptionKMSType": "vaulttenantsa",
      "vaultAddress": "<https://hostname_or_ip_of_vault_server:port>",
      "vaultTLSServerName": "<vault TLS server name>",
      "vaultAuthPath": "/v1/auth/kubernetes/login",
      "vaultAuthNamespace": "<vault auth namespace name>"
      "vaultNamespace": "<vault namespace name>",
      "vaultBackendPath": "<vault backend path name>",
      "vaultCAFromSecret": "<secret containing CA cert>",
      "vaultClientCertFromSecret": "<secret containing client cert>",
      "vaultClientCertKeyFromSecret": "<secret containing client private key>",
      "tenantSAName": "<service account name in the tenant namespace>"
    }
metadata:
  name: csi-kms-connection-details
```

Table 1. YAML values and descriptions

Value	Description
`encryptionKMSType`	Set to `vaulttenantsa` to use service accounts for authentication with vault.
`vaultAddress`	The hostname or IP address of the vault server with the port number.
`vaultTLSServerName`	(Optional) The vault TLS server name.
`vaultAuthPath`	(Optional) The path where Kubernetes auth method is enabled in Vault. The default path is kubernetes. If the auth method is enabled in a different path other than kubernetes, this variable needs to be set as "/v1/auth/<path>/login".
`vaultAuthNamespace`	(Optional) The Vault namespace where Kubernetes auth method is enabled.
`vaultNamespace`	(Optional) The Vault namespace where the backend path being used to store the keys exists.
`vaultBackendPath`	The backend path in Vault where the encryption keys are stored.
`vaultCAFromSecret`	The secret in the Fusion Data Foundation cluster containing the CA certificate from Vault.
`vaultClientCertFromSecret`	The secret in the Fusion Data Foundation cluster containing the client certificate from Vault.
`vaultClientCertKeyFromSecret`	The secret in the Fusion Data Foundation cluster containing the client private key from Vault.
`tenantSAName`	(Optional) The service account name in the tenant namespace. The default value is `ceph-csi-vault-sa`. If a different name is to be used, this variable has to be set accordingly.

Creating a storage class for persistent volume encryption

Create a storage class for persistent volume (PV) encryption.

Before you begin

Based on your use case, ensure that you configure access to the key management system (KMS) for one of the following options:

- Using `vaulttokens`: Ensure to configure access as described in Configuring access to KMS by using vaulttokens.
- Using `vaulttenantsa`: Ensure to configure access as described in Configuring access by using vaulttenantsa.
- Using Thales CipherTrust Manager (using KMIP): Ensure to configure access as described in Configuring access to KMS by using Thales CipherTrust Manager.
- (For users on Azure platform only) Using Azure Vault: Ensure to set up client authentication and fetch the client credentials from Azure using the following steps:

    1. Create Azure Vault. For more information, see Quickstart: Create a key vault using the Azure portal in Microsoft product documentation.
    2. Create Service Principal with certificate based authentication. For more information, see Create an Azure service principal with Azure CLI in Microsoft product documentation.
    3. Set Azure Key Vault role based access control (RBAC). For more information, see Enable Azure RBAC permissions on Key Vault in Microsoft product documentation.

Procedure

1. In the Red Hat® OpenShift® Container Platform web console, go to Storage > StorageClasses.
2. Enter the storage class Name and Description.
3. Select either Delete or Retain for the Reclaim Policy.
   Delete is selected by default.
4. Select either Immediate or WaitForConsumer for the Volume binding mode.
   WaitForConsumer is selected by default.
5. Select RBD Provisioner `openshift-storage.rbd.csi.ceph.com` which is the plugin used for provisioning the persistent volumes.
6. Select Storage Pool where the volume data is stored from the list or create a new pool.
7. Select the Enable encryption checkbox.
   Choose one of the following options to set the KMS connection details:

    - Choose existing KMS connection: Select an existing KMS connection from the drop-down list. The list is populated from the connection details available in the `csi-kms-connection-details` ConfigMap.
        - Select the Provider from the drop-down list.
        - Select the Key service for the given provider from the list.

- Create new KMS connection: This is applicable for `vaulttokens` and `Thales CipherTrust Manager (using KMIP)` only.
  - Select one of the following Key Management Service Provider and provide the required details.
    - Vault
      - Enter a unique Connection Name, host Address of the Vault server ('https://<hostname or ip>'), Port number and Token.
      - Expand Advanced Settings to enter additional settings and certificate details based on your `Vault` configuration:
        - Enter the Key Value secret path in Backend Path that is dedicated and unique to Fusion Data Foundation.
        - Optional: Enter TLS Server Name and Vault Enterprise Namespace.
        - Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate, and Client Private Key.
        - Click Save.
    - Thales CipherTrust Manager (using KMIP)
      - Enter a unique Connection Name.
      - In the Address and Port sections, enter the IP of Thales CipherTrust Manager and the port where the KMIP interface is enabled. For example, Address: 123.34.3.2, Port: 5696.
      - Upload the Client Certificate, CA certificate, and Client Private Key.
      - Enter the Unique Identifier for the key to be used for encryption and decryption, generated in the previous step.
      - The TLS Server field is optional and used when there is no DNS entry for the KMIP endpoint. For example, `kmip_all_<port>.ciphertrustmanager.local`.
    - Azure Key Vault (Only for Azure users on Azure platform)

      For information about setting up client authentication and fetching the client credentials, see the Prerequisites in Creating an OpenShift Data Foundation cluster section of the *Deploying OpenShift Data Foundation using Microsoft Azure* guide.
      - Enter a unique Connection name for the key management service within the project.
      - Enter Azure Vault URL.
      - Enter Client ID.
      - Enter Tenant ID.
      - Upload Certificate file in `.PEM` format and the certificate file must include a client certificate and a private key.
  - Click Save.
  - Click Create.
8. Edit the ConfigMap to add the `vaultBackend` parameter if the HashiCorp Vault setup does not allow automatic detection of the Key/Value (KV) secret engine API version used by the backend path.
   Note: `vaultBackend` is an optional parameters that is added to the ConfigMap to specify the version of the KV secret engine API associated with the backend path. Ensure that the value matches the KV secret engine API version that is set for the backend path, otherwise it might result in a failure during persistent volume claim (PVC) creation.
   a. Identify the encryptionKMSID being used by the newly created storage class.
      i. From the Red Hat OpenShift Container Platform web console, go to Storage > Storage Classes.
      ii. Click the storage class name and go to the YAML tab.
      iii. Capture the encryptionKMSID being used by the storage class.
      For example:

      ```
      encryptionKMSID: 1-vault
      ```

   b. Go to Workloads > ConfigMaps.
   c. View the KMS connection details, by clicking csi-kms-connection-details.
   d. Edit the ConfigMaps.
      i. Click Action menu > Edit ConfigMap.
      ii. Add the vaultBackend parameter depending on the backend that is configured for the previously identified `encryptionKMSID`.
      You can assign `kv` for KV secret engine API, version 1 and `kv-v2` for KV secret engine API, version 2.

      Example:

      ```
      kind: ConfigMap
      apiVersion: v1
      metadata:
        name: csi-kms-connection-details
      [...]
      data:
        1-vault: |-
          {
            "encryptionKMSType": "vaulttokens",
            "kmsServiceName": "1-vault",
            [...]
            "vaultBackend": "kv-v2"
          }
        2-vault: |-
          {
            "encryptionKMSType": "vaulttenantsa",
            [...]
            "vaultBackend": "kv"
          }
      ```

      iii. Click Save.

What to do next

The storage class can be used to create encrypted persistent volumes. For more information, see Managing persistent volume claims.

Important: IBM works with the technology partners to provide this documentation as a service to the customers. However, IBM does not provide support for the HashiCorp product. For technical assistance with this product, contact HashiCorp.

- **Overriding Vault connection details using tenant ConfigMap**
  The Vault connections details can be reconfigured per tenant by creating a ConfigMap in the Openshift namespace with configuration options that differ from the values set in the `csi-kms-connection-details` ConfigMap in the `openshift-storage` namespace. The ConfigMap needs to be located in the tenant

namespace. The values in the ConfigMap in the tenant namespace overrides the values set in the `csi-kms-connection-details` ConfigMap for the encrypted Persistent Volumes created in that namespace.

Overriding Vault connection details using tenant ConfigMap

The Vault connections details can be reconfigured per tenant by creating a ConfigMap in the Openshift namespace with configuration options that differ from the values set in the `csi-kms-connection-details` ConfigMap in the `openshift-storage` namespace. The ConfigMap needs to be located in the tenant namespace. The values in the ConfigMap in the tenant namespace overrides the values set in the `csi-kms-connection-details` ConfigMap for the encrypted Persistent Volumes created in that namespace.

Procedure

1. Ensure that you are in the tenant namespace.
2. Click Workloads > ConfigMaps > Create ConfigMap.
3. Edit the YAML file.
   The following is a sample YAML. The values to be overidden for the given tenant namespace can be specified under the `data` section as shown in the following:

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: ceph-csi-kms-config
data:
  vaultAddress: "<vault_address:port>"
  vaultBackendPath: "<backend_path>"
  vaultTLSServerName: "<vault_tls_server_name>"
  vaultNamespace: "<vault_namespace>"
```

4. Click Create.

Enabling and disabling key rotation when using KMS

Security common practices require periodic encryption of key rotation. You can enable or disable key rotation when using key management system (KMS).

- **Enforcing precedence for key rotation**
  Configure how Fusion Data Foundation determines key rotation order by enforcing precedence between persistent volume claims (PVCs) and storage classes.
- **Enabling key rotation**
  Enable key rotation by adding the required annotation to a `Namespace`, `StorageClass`, or `PersistentVolumeClaim`.
- **Disabling key rotation**
  Disable key rotation for a storage class or specific persistent volume claim (PVC) by updating annotations or modifying the `EncryptionKeyRotationCronJob` to stop scheduled rotation.

Enforcing precedence for key rotation

Configure how Fusion Data Foundation determines key rotation order by enforcing precedence between persistent volume claims (PVCs) and storage classes.

In key rotation, precedence refers to the order in which the system checks for scheduled annotations. In Fusion Data Foundation, the default precedence is set to storage class (recommended). This means the system reads annotations only from the storage class.

However, if you want the system to check the PVC first and then fall back to the storage class, you can configure this behavior by setting the `schedule-precedence` to PVC in the CSI-addons ConfigMap.

You can define the `ConfigMap` for csi-addons as follows:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: csi-addons-config
  namespace: openshift-storage
data:
  "schedule-precedence": "pvc"
```

Restart the `csi-addons` operator pod for the changes to take effect:

```
oc delete po -n openshift-storage -l "app.kubernetes.io/name=csi-addons"
```

Enabling key rotation

Enable key rotation by adding the required annotation to a `Namespace`, `StorageClass`, or `PersistentVolumeClaim`.

To enable key rotation, add the annotation `keyrotation.csiaddons.openshift.io/schedule: <value>` to `PersistentVolumeClaim`, `Namespace`, or `StorageClass` (in the decreasing order of precedence).

`<value>` can be `@hourly`, `@daily`, `@weekly`, `@monthly`, or `@yearly`. If `<value>`is empty, the default is `@weekly`. The following examples use `@weekly`.

Important: Key rotation is only supported for RBD backed volumes.

Annotating Namespace

Command example:

```
oc get namespace default
```

Example output:

```
NAME      STATUS   AGE
default   Active   5d2h
```

Command example:

```
oc annotate namespace default "keyrotation.csiaddons.openshift.io/schedule=@weekly"
```

Example output:

```
namespace/default annotated
```

Annotating StorageClass

Command example:

```
oc get storageclass rbd-sc
```

Example output:

```
NAME       PROVISIONER        RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
rbd-sc     rbd.csi.ceph.com   Delete          Immediate           true                   5d2h
```

Command example:

```
oc annotate storageclass rbd-sc "keyrotation.csiaddons.openshift.io/schedule=@weekly"
```

Example output:

```
storageclass.storage.k8s.io/rbd-sc annotated
```

Annotating PersistentVolumeClaim

Command example:

```
oc get pvc data-pvc
```

Example output:

```
NAME      STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS      AGE
data-pvc  Bound    pvc-f37b8582-4b04-4676-88dd-e1b95c6abf74   1Gi        RWO            default           20h
```

Command example:

```
oc annotate pvc data-pvc "keyrotation.csiaddons.openshift.io/schedule=@weekly"
```

Example output:

```
persistentvolumeclaim/data-pvc annotated
```

Command example:

```
oc get encryptionkeyrotationcronjobs.csiaddons.openshift.io
```

Example output:

```
NAME                  SCHEDULE    SUSPEND   ACTIVE   LASTSCHEDULE   AGE
data-pvc-1642663516   @weekly                                       3s
```

Command example:

```
oc annotate pvc data-pvc "keyrotation.csiaddons.openshift.io/schedule=*/1 * * * *" --overwrite=true
```

Example output:

```
persistentvolumeclaim/data-pvc annotated
```

Command example:

```
oc get encryptionkeyrotationcronjobs.csiaddons.openshift.io
```

Example output:

```
NAME                  SCHEDULE    SUSPEND   ACTIVE   LASTSCHEDULE   AGE
data-pvc-1642664617   */1 * * * *                                   3s
```

Disabling key rotation

Disable key rotation for a storage class or specific persistent volume claim (PVC) by updating annotations or modifying the **EncryptionKeyRotationCronJob** to stop scheduled rotation.

You can disable key rotation for the following:

- All the persistent volume claims (PVCs) of storage class
- A specific PVC

Disabling key rotation for all PVCs of a storage class

To disable key rotation for all PVCs, update the annotation of the storage class:

Command example:

```
oc get storageclass rbd-sc
```

Example output:

```
NAME       PROVISIONER        RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
rbd-sc     rbd.csi.ceph.com   Delete          Immediate           true                   5d2h
```

Command example:

```
oc annotate storageclass rbd-sc "keyrotation.csiaddons.openshift.io/enable: false"
```

Example output:

```
storageclass.storage.k8s.io/rbd-sc annotated
```

Disabling key rotation for a specific persistent volume claim

1. Identify the **EncryptionKeyRotationCronJob** CR for the PVC you want to disable key rotation on:

   ```
   oc get encryptionkeyrotationcronjob -o jsonpath='{range .items[?(@.spec.jobTemplate.spec.target.persistentVolumeClaim=="<PVC_NAME>")]}{.metadata.name}{"\n"}{end}'
   ```

   Where **<PVC_NAME>** is the name of the PVC that you want to disable.

2. Apply the following to the **EncryptionKeyRotationCronJob** CR from the previous step to disable the key rotation:
   a. Update the **csiaddons.openshift.io/state** annotation from **managed** to **unmanaged**:

      ```
      oc annotate encryptionkeyrotationcronjob <encryptionkeyrotationcronjob_name> "csiaddons.openshift.io/state=unmanaged" --overwrite=true
      ```

      Where **<encryptionkeyrotationcronjob_name>** is the name of the **EncryptionKeyRotationCronJob** CR.

   b. Add **suspend: true** under the **spec** field:

      ```
      oc patch encryptionkeyrotationcronjob <encryptionkeyrotationcronjob_name> -p '{"spec": {"suspend": true}}' --type=merge.
      ```

   c. Save and exit. The key rotation is disabled for the PVC.

Enabling and disabling encryption in-transit post deployment

You can enable encryption in-transit for the existing clusters after the deployment of clusters both in internal and external modes.

- **Enabling encryption in-transit after deployment in internal mode**
  Learn how to enable encryption in-transit after the deployment of a cluster in internal mode.
- **Disabling encryption in-transit after deployment in internal mode**
  Learn how to disable encryption in-transit after the deployment of a cluster in internal mode.
- **Enabling encryption in-transit after deployment in external mode**
  Learn how to enable encryption in-transit after the deployment of a cluster in external mode.
- **Disabling encryption in-transit after deployment in external mode**
  Learn how to disable encryption in-transit after the deployment of a cluster in external mode.

Enabling encryption in-transit after deployment in internal mode

Learn how to enable encryption in-transit after the deployment of a cluster in internal mode.

Before you begin

Fusion Data Foundation is deployed and a storage cluster is created.

Procedure

1. Patch the storagecluster to add encryption **enabled** as true to the storage cluster specification.

```
oc patch storagecluster ocs-storagecluster -n openshift-storage --type json --patch  '[{ "op": "replace", "path":
"/spec/network", "value": {"connections": {"encryption": {"enabled": true}}} }]'
storagecluster.ocs.openshift.io/ocs-storagecluster patched
```

2. Check the configurations.

```
oc get storagecluster ocs-storagecluster -n openshift-storage -o yaml | yq '.spec.network'
connections:
  encryption:
    enabled: true
```

3. Wait for around 10 minutes for Ceph daemons to restart and then check the pods.

```
oc get pods -n openshift-storage | grep rook-ceph
rook-ceph-crashcollector-ip-10-0-2-111.ec2.internal-796ffcm9kn9   1/1     Running     0          5m11s
rook-ceph-crashcollector-ip-10-0-27-61.ec2.internal-854b4d8sk5z   1/1     Running     0          5m9s
rook-ceph-crashcollector-ip-10-0-33-53.ec2.internal-589d9f4f8vx   1/1     Running     0          5m7s
rook-ceph-exporter-ip-10-0-2-111.ec2.internal-6d48cdc5fd-2tmsl    1/1     Running     0          5m9s
rook-ceph-exporter-ip-10-0-27-61.ec2.internal-546c66c7cc-9lnpz    1/1     Running     0          5m7s
rook-ceph-exporter-ip-10-0-33-53.ec2.internal-b5555994c-x8mzz     1/1     Running     0          5m5s
rook-ceph-mds-ocs-storagecluster-cephfilesystem-a-7bd754f6vwps2   2/2     Running     0          4m56s
rook-ceph-mds-ocs-storagecluster-cephfilesystem-b-6cc5cc647c78m   2/2     Running     0          4m30s
rook-ceph-mgr-a-6f8467578d-f8279                                  3/3     Running     0          3m40s
rook-ceph-mgr-b-66754d99cf-9q58g                                  3/3     Running     0          3m27s
rook-ceph-mon-a-75bc5dd655-tvdqf                                  2/2     Running     0          4m7s
rook-ceph-mon-b-6b6d4d9b4c-tjbpz                                  2/2     Running     0          4m55s
rook-ceph-mon-c-7456bb5f67-rtwpj                                  2/2     Running     0          4m32s
rook-ceph-operator-7b5b9cdb9b-tvmb6                               1/1     Running     0          45m
rook-ceph-osd-0-b78dd99f6-n4wbm                                   2/2     Running     0          3m3s
rook-ceph-osd-1-5887bf6d8d-2sncc                                  2/2     Running     0          2m39s
rook-ceph-osd-2-784b59c4c8-44phh                                  2/2     Running     0          2m14s
rook-ceph-osd-prepare-a075cf185c9b2e5d92ec3f7769565e38-ztrms      0/1     Completed   0          42m
rook-ceph-osd-prepare-b4b48dc5e3bef99ab377e2a255a9142a-mvgnd      0/1     Completed   0          42m
rook-ceph-osd-prepare-fae2ea2ad4aacbf62010ae5b60b87f57-6t9l5      0/1     Completed   0          42m

oc get storagecluster -n openshift-storage
NAME                 AGE   PHASE   EXTERNAL   CREATED AT             VERSION
ocs-storagecluster   27m   Ready              2024-11-06T16:15:26Z   4.18.0
```

4. Remount existing volumes.
   Depending on your best practices for application maintenance, you can choose the best approach for your environment to remount or remap volumes. One way to remount is to delete the existing application pod and bring up another application pod to use the volume. Another option is to drain the nodes where the applications are running. This ensures that the volume is unmounted from the current pod and then mounted to a new pod, allowing for remapping or remounting of the volume.

Disabling encryption in-transit after deployment in internal mode

Learn how to disable encryption in-transit after the deployment of a cluster in internal mode.

Before you begin

- Fusion Data Foundation is deployed and a storage cluster is created.
- Encryption in-transit is enabled.

Procedure

1. Patch the storagecluster to update encryption `enabled` as `false` in the storage cluster specification.

```
oc patch storagecluster ocs-storagecluster -n openshift-storage --type json --patch  '[{ "op": "replace", "path":
"/spec/network", "value": {"connections": {"encryption": {"enabled": false}}} }]'
storagecluster.ocs.openshift.io/ocs-storagecluster patched
```

2. Check the configurations.

```
oc get storagecluster ocs-storagecluster -n openshift-storage -o yaml | yq '.spec.network'

connections:
  encryption:
    enabled: false
```

3. Wait for around 10 minutes for Ceph daemons to restart and then check the pods.

```
oc get pods -n openshift-storage | grep rook-ceph
rook-ceph-crashcollector-ip-10-0-2-111.ec2.internal-796ffcm9kn9   1/1     Running     0          5m11s
rook-ceph-crashcollector-ip-10-0-27-61.ec2.internal-854b4d8sk5z   1/1     Running     0          5m9s
rook-ceph-crashcollector-ip-10-0-33-53.ec2.internal-589d9f4f8vx   1/1     Running     0          5m7s
rook-ceph-exporter-ip-10-0-2-111.ec2.internal-6d48cdc5fd-2tmsl    1/1     Running     0          5m9s
rook-ceph-exporter-ip-10-0-27-61.ec2.internal-546c66c7cc-9lnpz    1/1     Running     0          5m7s
rook-ceph-exporter-ip-10-0-33-53.ec2.internal-b5555994c-x8mzz     1/1     Running     0          5m5s
rook-ceph-mds-ocs-storagecluster-cephfilesystem-a-7bd754f6vwps2   2/2     Running     0          4m56s
rook-ceph-mds-ocs-storagecluster-cephfilesystem-b-6cc5cc647c78m   2/2     Running     0          4m30s
rook-ceph-mgr-a-6f8467578d-f8279                                  3/3     Running     0          3m40s
rook-ceph-mgr-b-66754d99cf-9q58g                                  3/3     Running     0          3m27s
rook-ceph-mon-a-75bc5dd655-tvdqf                                  2/2     Running     0          4m7s
rook-ceph-mon-b-6b6d4d9b4c-tjbpz                                  2/2     Running     0          4m55s
rook-ceph-mon-c-7456bb5f67-rtwpj                                  2/2     Running     0          4m32s
rook-ceph-operator-7b5b9cdb9b-tvmb6                               1/1     Running     0          45m
```

```
rook-ceph-osd-0-b78dd99f6-n4wbm                                   2/2     Running     0          3m3s
rook-ceph-osd-1-5887bf6d8d-2sncc                                  2/2     Running     0          2m39s
rook-ceph-osd-2-784b59c4c8-44phh                                  2/2     Running     0          2m14s
rook-ceph-osd-prepare-a075cf185c9b2e5d92ec3f7769565e38-ztrms      0/1     Completed   0          42m
rook-ceph-osd-prepare-b4b48dc5e3bef99ab377e2a255a9142a-mvgnd      0/1     Completed   0          42m
rook-ceph-osd-prepare-fae2ea2ad4aacbf62010ae5b60b87f57-6t9l5      0/1     Completed   0          42m
```

```
oc get storagecluster -n openshift-storage
NAME                 AGE   PHASE   EXTERNAL   CREATED AT             VERSION
ocs-storagecluster   27m   Ready              2024-11-06T16:15:26Z   4.18.0
```

4. Remount existing volumes.
   Depending on your best practices for application maintenance, you can choose the best approach for your environment to remount or remap volumes. One way to remount is to delete the existing application pod and bring up another application pod to use the volume. Another option is to drain the nodes where the applications are running. This ensures that the volume is unmounted from the current pod and then mounted to a new pod, allowing for remapping or remounting of the volume.

Enabling encryption in-transit after deployment in external mode

Learn how to enable encryption in-transit after the deployment of a cluster in external mode.

Before you begin

Fusion Data Foundation is deployed and a storage cluster is created.

Procedure

1. Patch the storagecluster to add encryption `enabled` as true to the storage cluster specification.

```
oc patch storagecluster ocs-external-storagecluster -n openshift-storage --type json --patch  '[{ "op": "replace", "path":
"/spec/network", "value": {"connections": {"encryption": {"enabled": true}}} }]'
storagecluster.ocs.openshift.io/ocs-external-storagecluster patched
```

2. Check the connection settings in the CR.

```
oc get storagecluster
NAME                          AGE   PHASE   EXTERNAL   CREATED AT             VERSION
ocs-external-storagecluster   9h    Ready   true       2024-11-06T20:48:03Z   4.18.0

oc get storagecluster ocs-external-storagecluster -o yaml | yq '.spec.network.connections'
encryption:
  enabled: true
```

- **Applying encryption in-transit on Red Hat Ceph storage cluster**
  Learn how to apply encryption in-transit on Red Hat Ceph storage cluster.
- **Remount existing volumes**
  Remounting or remapping an existing volume can be accomplished by deleting the application pod to bring up a new one, or by draining the node where the application is running to unmount and remount the volume on a new pod.

Applying encryption in-transit on Red Hat Ceph storage cluster

Learn how to apply encryption in-transit on Red Hat Ceph storage cluster.

Procedure

1. Apply Encryption in-transit settings.

```
root@ceph-client ~]# ceph config set global ms_client_mode secure
[root@ceph-client ~]# ceph config set global ms_cluster_mode secure
[root@ceph-client ~]# ceph config set global ms_service_mode secure
[root@ceph-client ~]# ceph config set global rbd_default_map_options ms_mode=secure
```

2. Check the settings.

```
[root@ceph-client ~]# ceph config dump | grep ms_
ceph config dump | grep ms_
global   basic      ms_client_mode secure                    *
global   basic      ms_cluster_mode secure                   *
global   basic      ms_service_mode secure                   *
global   advanced   rbd_default_map_options ms_mode=secure   *
```

3. Restart all Ceph daemons.

```
[root@ceph-client ~]# ceph orch ls --format plain | tail -n +2 | awk '{print $1}' | xargs -I {} ceph orch restart {}
Scheduled to restart alertmanager.osd-0 on host 'osd-0'
Scheduled to restart ceph-exporter.osd-0 on host 'osd-0'
Scheduled to restart ceph-exporter.osd-2 on host 'osd-2'
Scheduled to restart ceph-exporter.osd-3 on host 'osd-3'
Scheduled to restart ceph-exporter.osd-1 on host 'osd-1'
Scheduled to restart crash.osd-0 on host 'osd-0'
Scheduled to restart crash.osd-2 on host 'osd-2'
```

```
Scheduled to restart crash.osd-3 on host 'osd-3'
Scheduled to restart crash.osd-1 on host 'osd-1'
Scheduled to restart grafana.osd-0 on host 'osd-0'
Scheduled to restart mds.fsvol001.osd-0.lpciqk on host 'osd-0'
Scheduled to restart mds.fsvol001.osd-2.wocnxz on host 'osd-2'
Scheduled to restart mgr.osd-0.dtkyni on host 'osd-0'
Scheduled to restart mgr.osd-2.kqcxwu on host 'osd-2'
Scheduled to restart mon.osd-2 on host 'osd-2'
Scheduled to restart mon.osd-3 on host 'osd-3'
Scheduled to restart mon.osd-1 on host 'osd-1'
Scheduled to restart node-exporter.osd-0 on host 'osd-0'
Scheduled to restart node-exporter.osd-2 on host 'osd-2'
Scheduled to restart node-exporter.osd-3 on host 'osd-3'
Scheduled to restart node-exporter.osd-1 on host 'osd-1'
Scheduled to restart osd.1 on host 'osd-0'
Scheduled to restart osd.4 on host 'osd-0'
Scheduled to restart osd.0 on host 'osd-2'
Scheduled to restart osd.5 on host 'osd-2'
Scheduled to restart osd.2 on host 'osd-3'
Scheduled to restart osd.6 on host 'osd-3'
Scheduled to restart osd.3 on host 'osd-1'
Scheduled to restart osd.7 on host 'osd-1'
Scheduled to restart prometheus.osd-0 on host 'osd-0'
Scheduled to restart rgw.rgw.ssl.osd-1.smzpfj on host 'osd-1'
```

Wait for the restarting of all the daemons.

Remount existing volumes

Remounting or remapping an existing volume can be accomplished by deleting the application pod to bring up a new one, or by draining the node where the application is running to unmount and remount the volume on a new pod.

Depending on your best practices for application maintenance, you can choose the best approach for your environment to remount or remap volumes. One way to remount is to delete the existing application pod and bring up another application pod to use the volume. Another option is to drain the nodes where the applications are running. This ensures that the volume is unmounted from the current pod and then mounted to a new pod, allowing for remapping or remounting of the volume.

Disabling encryption in-transit after deployment in external mode

Learn how to disable encryption in-transit after the deployment of a cluster in external mode.

Before you begin

- Fusion Data Foundation is deployed and a storage cluster is created.
- Encryption in-transit is enabled for the external mode cluster.

Procedure

1. Remove and check encryption in-transit configurations.

```
[root@ceph-client ~]# ceph config rm global ms_client_mode
[root@ceph-client ~]# ceph config rm global ms_cluster_mode
[root@ceph-client ~]# ceph config rm global ms_service_mode
[root@ceph-client ~]# ceph config rm global rbd_default_map_options

[root@ceph-client ~]# ceph config dump | grep ms_
[root@ceph-client ~]#
```

2. Restart all Ceph daemons.

```
[root@ceph-client ~]# ceph orch ls --format plain | tail -n +2 | awk '{print $1}' | xargs -I {} ceph orch restart {}
Scheduled to restart alertmanager.osd-0 on host 'osd-0'
Scheduled to restart ceph-exporter.osd-0 on host 'osd-0'
Scheduled to restart ceph-exporter.osd-2 on host 'osd-2'
Scheduled to restart ceph-exporter.osd-3 on host 'osd-3'
Scheduled to restart ceph-exporter.osd-1 on host 'osd-1'
Scheduled to restart crash.osd-0 on host 'osd-0'
Scheduled to restart crash.osd-2 on host 'osd-2'
Scheduled to restart crash.osd-3 on host 'osd-3'
Scheduled to restart crash.osd-1 on host 'osd-1'
Scheduled to restart grafana.osd-0 on host 'osd-0'
Scheduled to restart mds.fsvol001.osd-0.lpciqk on host 'osd-0'
Scheduled to restart mds.fsvol001.osd-2.wocnxz on host 'osd-2'
Scheduled to restart mgr.osd-0.dtkyni on host 'osd-0'
Scheduled to restart mgr.osd-2.kqcxwu on host 'osd-2'
Scheduled to restart mon.osd-2 on host 'osd-2'
Scheduled to restart mon.osd-3 on host 'osd-3'
Scheduled to restart mon.osd-1 on host 'osd-1'
Scheduled to restart node-exporter.osd-0 on host 'osd-0'
Scheduled to restart node-exporter.osd-2 on host 'osd-2'
Scheduled to restart node-exporter.osd-3 on host 'osd-3'
Scheduled to restart node-exporter.osd-1 on host 'osd-1'
Scheduled to restart osd.1 on host 'osd-0'
Scheduled to restart osd.4 on host 'osd-0'
```

```
Scheduled to restart osd.0 on host 'osd-2'
Scheduled to restart osd.5 on host 'osd-2'
Scheduled to restart osd.2 on host 'osd-3'
Scheduled to restart osd.6 on host 'osd-3'
Scheduled to restart osd.3 on host 'osd-1'
Scheduled to restart osd.7 on host 'osd-1'
Scheduled to restart prometheus.osd-0 on host 'osd-0'
Scheduled to restart rgw.rgw.ssl.osd-1.smzpfj on host 'osd-1'

[root@ceph-client ~]# ceph orch ps
NAME                          HOST   PORTS              STATUS           REFRESHED  AGE  MEM USE  MEM LIM  VERSION
IMAGE ID      CONTAINER ID
alertmanager.osd-0            osd-0  *:9093,9094        running (116s)     9s ago  10h    19.5M        -  0.26.0
7dbf12091920  4694a72d4bbd
ceph-exporter.osd-0           osd-0                     running (19s)      9s ago  10h    7310k        -  18.2.1-229.el9cp
3fd804e38f5b  49bdc7d99471
ceph-exporter.osd-1           osd-1                     running (97s)     26s ago  10h    7285k        -  18.2.1-229.el9cp
3fd804e38f5b  7000d59d23b4
ceph-exporter.osd-2           osd-2                     running (76s)     26s ago  10h    7306k        -  18.2.1-229.el9cp
3fd804e38f5b  3907515cc352
ceph-exporter.osd-3           osd-3                     running (49s)     26s ago  10h    6971k        -  18.2.1-229.el9cp
3fd804e38f5b  3f3952490780
crash.osd-0                   osd-0                     running (17s)      9s ago  10h    6878k        -  18.2.1-229.el9cp
3fd804e38f5b  38e041fb86e3
crash.osd-1                   osd-1                     running (96s)     26s ago  10h    6895k        -  18.2.1-229.el9cp
3fd804e38f5b  21ce3ef7d896
crash.osd-2                   osd-2                     running (74s)     26s ago  10h    6899k        -  18.2.1-229.el9cp
3fd804e38f5b  210ca9c8d928
crash.osd-3                   osd-3                     running (47s)     26s ago  10h    6899k        -  18.2.1-229.el9cp
3fd804e38f5b  710d42d9d138
grafana.osd-0                 osd-0  *:3000             running (114s)     9s ago  10h    72.9M        -  10.4.0-pre
f142b583a1b1  3dc5e2248e95
mds.fsvol001.osd-0.qjntcu     osd-0                     running (99s)      9s ago  10h    17.5M        -  18.2.1-229.el9cp
3fd804e38f5b  50efa881c04b
mds.fsvol001.osd-2.qneujv     osd-2                     running (51s)     26s ago  10h    15.3M        -  18.2.1-229.el9cp
3fd804e38f5b  a306f2d2d676
mgr.osd-0.zukgyq              osd-0  *:9283,8765,8443   running (21s)      9s ago  10h     442M        -  18.2.1-229.el9cp
3fd804e38f5b  8ef9b728675e
mgr.osd-1.jqfyal              osd-1  *:8443,9283,8765   running (92s)     26s ago  10h     480M        -  18.2.1-229.el9cp
3fd804e38f5b  1ab52db89bfd
mon.osd-1                     osd-1                     running (90s)     26s ago  10h    41.7M    2048M  18.2.1-229.el9cp
3fd804e38f5b  88d1fe1e10ac
mon.osd-2                     osd-2                     running (72s)     26s ago  10h    31.1M    2048M  18.2.1-229.el9cp
3fd804e38f5b  02f57d3bb44f
mon.osd-3                     osd-3                     running (45s)     26s ago  10h    24.0M    2048M  18.2.1-229.el9cp
3fd804e38f5b  5e3783f2b4fa
node-exporter.osd-0           osd-0  *:9100             running (15s)      9s ago  10h    7843k        -  1.7.0
8c904aa522d0  2dae2127349b
node-exporter.osd-1           osd-1  *:9100             running (94s)     26s ago  10h    11.2M        -  1.7.0
8c904aa522d0  010c3fcd55cd
node-exporter.osd-2           osd-2  *:9100             running (69s)     26s ago  10h    17.2M        -  1.7.0
8c904aa522d0  436f2d513f31
node-exporter.osd-3           osd-3  *:9100             running (41s)     26s ago  10h    12.4M        -  1.7.0
8c904aa522d0  5579f0d494b8
osd.0                         osd-0                     running (109s)     9s ago  10h     126M    4096M  18.2.1-229.el9cp
3fd804e38f5b  997076cd39d4
osd.1                         osd-1                     running (85s)     26s ago  10h     139M    4096M  18.2.1-229.el9cp
3fd804e38f5b  08b720f0587d
osd.2                         osd-2                     running (65s)     26s ago  10h     143M    4096M  18.2.1-229.el9cp
3fd804e38f5b  104ad4227163
osd.3                         osd-3                     running (36s)     26s ago  10h    94.5M    1435M  18.2.1-229.el9cp
3fd804e38f5b  db8b265d9f43
osd.4                         osd-0                     running (104s)     9s ago  10h     164M    4096M  18.2.1-229.el9cp
3fd804e38f5b  50dcbbf7e012
osd.5                         osd-1                     running (80s)     26s ago  10h     131M    4096M  18.2.1-229.el9cp
3fd804e38f5b  63b21fe970b5
osd.6                         osd-3                     running (32s)     26s ago  10h     243M    1435M  18.2.1-229.el9cp
3fd804e38f5b  26c7ba208489
osd.7                         osd-2                     running (61s)     26s ago  10h     130M    4096M  18.2.1-229.el9cp
3fd804e38f5b  871a2b75e64f
prometheus.osd-0              osd-0  *:9095             running (12s)      9s ago  10h    44.6M        -  2.48.0
58069186198d  e49a064d2478
rgw.rgw.ssl.osd-1.bsmbgd      osd-1  *:80               running (78s)     26s ago  10h    75.4M        -  18.2.1-229.el9cp
3fd804e38f5b  d03c9f7ae4a4
```

3. Patch the storagecluster to update encryption `enabled` as `false` in the storage cluster specification.

```
oc patch storagecluster ocs-external-storagecluster -n openshift-storage --type json --patch  '[{ "op": "replace", "path":
"/spec/network", "value": {"connections": {"encryption": {"enabled": false}}} }]'
storagecluster.ocs.openshift.io/ocs-external-storagecluster patched
```

4. Check the configurations.

```
oc get storagecluster
NAME                          AGE   PHASE   EXTERNAL   CREATED AT             VERSION
ocs-external-storagecluster   12h   Ready   true       2024-11-06T20:48:03Z   4.18.0

oc get storagecluster ocs-external-storagecluster -o yaml | yq '.spec.network.connections'
encryption:
  enabled: false
```

5. Remount existing volumes.
   Depending on your best practices for application maintenance, you can choose the best approach for your environment to remount or remap volumes. One way to remount is to delete the existing application pod and bring up another application pod to use the volume. Another option is to drain the nodes where the

applications are running. This ensures that the volume is unmounted from the current pod and then mounted to a new pod, allowing for remapping or remounting of the volume.

Block pools

This section provides you with information about how to configure block pools in internal mode.

The Fusion Data Foundation operator installs a default set of storage pools depending on the platform in use. These default storage pools are owned and controlled by the operator and it cannot be deleted or modified. With OpenShift Container Platform, you can create multiple custom storage pools which map to storage classes that provide the following features:

- Enable applications with their own high availability to use persistent volumes with two replicas, potentially improving application performance.
- Save space for persistent volume claims using storage classes with compression enabled.

Note: Multiple block pools are not supported for *external mode* Fusion Data Foundation clusters.

- **Block pools in internal mode**
  This section provides you with information on how to create, update, and delete block pools in Fusion Data Foundation internal mode.

Block pools in internal mode

This section provides you with information on how to create, update, and delete block pools in Fusion Data Foundation internal mode.

You can create custom storage pools which map to storage classes in internal mode.

- **Creating a block pool**
  Use this procedure to create custom storage pools which map to storage classes.
- **Updating an existing pool**
  Use this procedure to update existing storage pools.
- **Deleting a pool**
  Use this procedure to delete a pool in Fusion Data Foundation.

Creating a block pool

Use this procedure to create custom storage pools which map to storage classes.

Before you begin

Ensure that you have logged into the OpenShift Container Platform web console as an administrator.

Procedure

1. Go to Storage > Data Foundation.
2. In the Data Foundation window, click Storage Systems.
3. In the Storage systems tab, select the required storage system.
4. Go to the Storage pools tab and click Create storage pool.
5. In the "Create storage pool" dialog, do as follows:
   a. Select the Volume type as Block.
   b. Enter a Pool name.
   c. Select the Data protection policy as either 2-way Replication or 3-way Replication.
      Remember: The default pool does not support the 2-way replication data protection policy. However, you can use 2-way replication if you create an additional pool.
   d. Optional: Select the Data compression checkbox to optimize storage efficiency.
      Important: Enabling compression can impact application performance and might prove ineffective when data to be written is already compressed or encrypted. The application does not compress the data written before enabling compression.
   e. Click Create.

Updating an existing pool

Use this procedure to update existing storage pools.

Before you begin

Ensure that you have logged into the OpenShift Container Platform web console as an administrator.

Procedure

1. Go to Storage > Data Foundation.

2. In the Data Foundation window, click Storage Systems.
3. On the Storage systems tab, select the required storage system.
4. Go to the Storage pools tab.
5. On the Storage pools tab, click the ellipsis menu next to the pool you want to update and click Edit Pool.
6. In the Edit Storage Pool dialog, do as follows:
    a. Change the Data protection policy to either 2-way Replication or 3-way Replication.
    Remember: The default pool does not support the 2-way replication data protection policy. However, you can use 2-way replication if you create an additional pool.
    b. Optional: Change the data compression by selecting or deselecting the Data compression checkbox.
    Important: Enabling compression can impact application performance and might prove ineffective when data to be written is already compressed or encrypted. The application does not compress the data written before enabling compression.
    c. Click Save.

Deleting a pool

Use this procedure to delete a pool in Fusion Data Foundation.

Before you begin

Ensure that you have logged into the OpenShift Container Platform web console as an administrator.
Important: A pool cannot be deleted when it is bound to a PersistentVolumeClaim (PVC). You must detach all the resources before performing the deletion activity.

Procedure

1. Go to Storage > Data Foundation.
2. In the Data Foundation window, click Storage Systems.
3. On the Storage systems tab, select the required storage system.
4. Go to the Storage pools tab.
5. On the Storage pools tab, click the three-dot icon next to the pool you want to delete and click Delete Pool.
6. Click Delete to confirm the removal of the Pool.
    Remember: When a pool is deleted, the underlying Ceph pool is not deleted.

Configure storage for OpenShift Container Platform services

Use Fusion Data Foundation for core OpenShift Container Platform services, such as image registry, monitoring, and logging.

The process for configuring storage for these services depends on the infrastructure used in your Fusion Data Foundation deployment.

Warning: Always ensure that you have plenty of storage capacity for these services. If the storage for these critical services runs out of space, the cluster becomes inoperable and very difficult to recover.
IBM recommends configuring shorter curation and retention intervals for these services. For more information, see Observability > Monitoring > About OpenShift Container Platform monitoring > Monitoring stack for Red Hat OpenShift section in Red Hat OpenShift Container Platform product documentation for *Modifying retention time for Prometheus metrics data*.

If you do run out of storage space for these services, contact IBM Support.

- **Configuring Image Registry to use Fusion Data Foundation**
  OpenShift Container Platform provides a built in Container Image Registry which runs as a standard workload on the cluster. A registry is typically used as a publication target for images built on the cluster as well as a source of images for workloads running on the cluster.
- **Using Multicloud Object Gateway as OpenShift Image Registry backend storage**
  Use this section to use Multicloud Object Gateway (MCG) as OpenShift Container Platform (OCP) Image Registry backend storage in an on-prem OpenShift deployment.
- **Configuring monitoring to use Fusion Data Foundation**
  Fusion Data Foundation provides a monitoring stack that comprises of Prometheus and Alert Manager.
- **Overprovision level policy control [Technology Preview]**
  Overprovision control is a mechanism that enables you to define a quota on the amount of Persistent Volume Claims (PVCs) consumed from a storage cluster, based on the specific application namespace.
- **Cluster logging for Fusion Data Foundation**
  You can deploy cluster logging to aggregate logs for a range of OpenShift Container Platform services.

Configuring Image Registry to use Fusion Data Foundation

OpenShift Container Platform provides a built in Container Image Registry which runs as a standard workload on the cluster. A registry is typically used as a publication target for images built on the cluster as well as a source of images for workloads running on the cluster.

About this task

Warning: This process does not migrate data from an existing image registry to the new image registry. If you already have container images in your existing registry, back up your registry before you complete this process, and re-register your images when this process is complete.

Before you begin

Be sure that you have the following:

- You have administrative access to OpenShift Web Console.
- Fusion Data Foundation Operator is installed and running in the `openshift-storage` namespace. In OpenShift Web Console, go to Operators > Installed Operators to view installed operators.
- Image Registry Operator is installed and running in the `openshift-image-registry` namespace. In OpenShift Web Console, go to Administration > Cluster Settings > Cluster Operators to view cluster operators.
- A storage class with provisioner `openshift-storage.cephfs.csi.ceph.com` available. In OpenShift Web Console, go to Storage > StorageClasses to view available storage classes.

Procedure

Use the OpenShift Web Console to perform these steps.

1. Create a Persistent Volume Claim for the Image Registry to use.
   a. Go to Storage > Persistent Volume Claims.
   b. Set Project to `openshift-image-registry`.
   c. Click Create Persistent Volume Claims and fill out the form.
      - From the list of available storage classes retrieved in the previous step, specify the Storage Class with the provisioner `openshift-storage.cephfs.csi.ceph.com`.
      - Specify the Persistent Volume Claim Name, for example, `ocs4registry`.
      - Specify an Access Mode of Shared Access (RWX).
      - Specify a Size of at least 100 GB.
      - Click Create.

   Wait until the status of the new Persistent Volume Claim is listed as *Bound*.
2. Configure the cluster's Image Registry to use the new Persistent Volume Claim.
   a. Go to Administration > Custom Resource Definitions.
   b. Click the `Config` custom resource definition associated with the `imageregistry.operator.openshift.io` group.
   c. From the Instances tab, beside the cluster instance, go to Action Menu > Edit Config.
   d. Add the new Persistent Volume Claim as persistent storage for the Image Registry.
      i. If necessary, add the following under spec, replacing the existing storage section:

      ```
      storage:
        pvc:
          claim: <new-pvc-name>
      ```

      For example:

      ```
      storage:
        pvc:
          claim: ocs4registry
      ```

      ii. Click Save.
3. Verify that the new configuration is being used.
   a. Go to Workloads > Pods.
   b. Set Project to `openshift-image-registry`.
   c. Verify that the new `image-registry-*` pod appears with a status of *Running* and that the previous `image-registry-*` pod terminates.
   d. Click the new `image-registry-*` pod to view pod details.
   e. From Volumes verify that the `registry-storage` volume has a Type that matches your new Persistent Volume Claim. For example, `ocs4registry`.

Using Multicloud Object Gateway as OpenShift Image Registry backend storage

Use this section to use Multicloud Object Gateway (MCG) as OpenShift Container Platform (OCP) Image Registry backend storage in an on-prem OpenShift deployment.

About this task

You can use Multicloud Object Gateway (MCG) as OpenShift Container Platform (OCP) Image Registry backend storage in an on-prem OpenShift deployment.

Before you begin

Be sure that you have the following:

- You have administrative access to OpenShift Web Console.
- A running Fusion Data Foundation cluster with MCG.

Procedure

1. Create `ObjectBucketClaim` by following the steps in Dynamic Object Bucket Claim.
2. Create an `image-registry-private-configuration-user` secret.
   a. Go to the OpenShift web-console.
   b. Click **ObjectBucketClaim** -→ **ObjectBucketClaim Data**.
   c. In the **ObjectBucketClaim data**, look for `MCG access key` and `MCG secret key` in the `openshift-image-registry namespace`.
   d. Create the secret using the following command:

```
$ oc create secret generic image-registry-private-configuration-user --from-literal=REGISTRY_STORAGE_S3_ACCESSKEY=
<MCG Accesskey> --from-literal=REGISTRY_STORAGE_S3_SECRETKEY=<MCG Secretkey> --namespace openshift-image-registry
```

3. Change the status of `managementState` of Image Registry Operator to **Managed**.

```
$ oc patch configs.imageregistry.operator.openshift.io/cluster --type merge -p '{"spec": {"managementState": "Managed"}}'
```

4. Edit the `spec.storage` section of Image Registry Operator configuration file:
   a. Get the `unique-bucket-name` and `regionEndpoint` under the **Object Bucket Claim Data** section from the Web Console **OR** you can also get the information on regionEndpoint and unique-bucket-name from the command:

```
$ oc describe  noobaa
```

   b. Add `regionEndpoint` as `http://<Endpoint-name>:<port>` if the
      - storageclass is `ceph-rgw` storageclass and the
      - endpoint points to the internal SVC from the `openshift-storage` namespace.
   c. An `image-registry` pod spawns after you make the changes to the Operator registry configuration file.

```
$ oc edit configs.imageregistry.operator.openshift.io -n openshift-image-registry apiVersion:
imageregistry.operator.openshift.io/v1 kind: Config metadata: [..] name: cluster spec: [..] storage: s3: bucket: <Unique-
bucket-name> region: us-east-1 (Use this region as default) regionEndpoint: https://<Endpoint-name>:<port>
virtualHostedStyle: false
```

5. Reset the image registry settings to default.

```
$ oc get pods -n openshift-image-registry
```

What to do next

- Run the following command to check if you have configured the MCG as OpenShift Image Registry backend storage successfully.

```
$ oc get pods -n openshift-image-registry

NAME                                               READY   STATUS      RESTARTS    AGE

cluster-image-registry-operator-56d78bc5fb-bxcgv   2/2     Running           0         44d
image-pruner-1605830400-29r7k                      0/1     Completed   0         10h
image-registry-b6c8f4596-ln88h                     1/1     Running           0         17d
node-ca-2nxvz                                      1/1     Running       0         44d
node-ca-dtwjd                                      1/1     Running           0         44d
node-ca-h92rj                                      1/1     Running           0         44d
node-ca-k9bkd                                      1/1     Running           0         44d
node-ca-stkzc                                      1/1     Running           0         44d
node-ca-xn8h4                                      1/1     Running           0         44d
```

- (Optional) You can the run the following command to verify if you have configured the MCG as OpenShift Image Registry backend storage successfully.

```
$ oc describe pod image-registry-b6c8f4596-ln88h

Environment:

      REGISTRY_STORAGE_S3_REGIONENDPOINT:       http://s3.openshift-storage.svc

      REGISTRY_STORAGE:                         s3

      REGISTRY_STORAGE_S3_BUCKET:               bucket-registry-mcg

      REGISTRY_STORAGE_S3_REGION:               us-east-1

      REGISTRY_STORAGE_S3_ENCRYPT:              true

      REGISTRY_STORAGE_S3_VIRTUALHOSTEDSTYLE:   false

      REGISTRY_STORAGE_S3_USEDUALSTACK:         true

      REGISTRY_STORAGE_S3_ACCESSKEY:            <set to the key 'REGISTRY_STORAGE_S3_ACCESSKEY' in secret 'image-registry-
private-configuration'> Optional: false

      REGISTRY_STORAGE_S3_SECRETKEY:            <set to the key 'REGISTRY_STORAGE_S3_SECRETKEY' in secret 'image-registry-
private-configuration'> Optional: false

      REGISTRY_HTTP_ADDR:                       :5000

      REGISTRY_HTTP_NET:                        tcp

      REGISTRY_HTTP_SECRET:
57b943f691c878e342bac34e657b702bd6ca5488d51f839fecafa918a79a5fc6ed70184cab047601403c1f383e54d458744062dcaaa483816d82408bb5
6e686f

      REGISTRY_LOG_LEVEL:                       info

      REGISTRY_OPENSHIFT_QUOTA_ENABLED:         true

      REGISTRY_STORAGE_CACHE_BLOBDESCRIPTOR:    inmemory

      REGISTRY_STORAGE_DELETE_ENABLED:          true

      REGISTRY_OPENSHIFT_METRICS_ENABLED:       true

      REGISTRY_OPENSHIFT_SERVER_ADDR:           image-registry.openshift-image-registry.svc:5000

      REGISTRY_HTTP_TLS_CERTIFICATE:            /etc/secrets/tls.crt
```

```
      REGISTRY_HTTP_TLS_KEY:                  /etc/secrets/tls.key
```

Configuring monitoring to use Fusion Data Foundation

Fusion Data Foundation provides a monitoring stack that comprises of Prometheus and Alert Manager.

About this task

Important: Monitoring does not function if it runs out of storage space. Always ensure that you have plenty of storage capacity for monitoring.
IBM recommends configuring a short retention interval for this service. For more information, see Observability > Monitoring > About OpenShift Container Platform monitoring > Monitoring stack for Red Hat OpenShift section in Red Hat OpenShift Container Platform product documentation for *Modifying retention time for Prometheus metrics data*.

Before you begin

Be sure that you have the following:

- You have administrative access to OpenShift Web Console.
- Fusion Data Foundation Operator is installed and running in the `openshift-storage` namespace. In OpenShift Web Console, go to Operators > Installed Operators to view installed operators.
- Monitoring Operator is installed and running in the `openshift-monitoring` namespace. In OpenShift Web Console, go to Administration > Cluster Settings > Cluster Operators to view cluster operators.
- A storage class with provisioner `openshift-storage.rbd.csi.ceph.com` available. In OpenShift Web Console, go to Storage > StorageClasses to view available storage classes.

Procedure

Use the OpenShift Web Console to perform the following steps to configure Fusion Data Foundation as storage for the monitoring stack.

1. Go to Workloads > Config Maps.
2. Set the Project drop-down to openshift-monitoring.
3. Click Create Config Map and edit the form.
   a. Define a new `cluster-monitoring-config` Config Map.
   Replace the *<variables>* with your own values. For example, `retention: 24h` or `storage: 40Gi`.
   Replace the *storageClassName* with the `storageclass` that uses the provisioner `openshift-storage.rbd.csi.ceph.com`. In the following example the name of the *storageclass* is `ocs-storagecluster-ceph-rbd`.

   Example `cluster-monitoring-config` Config Map:

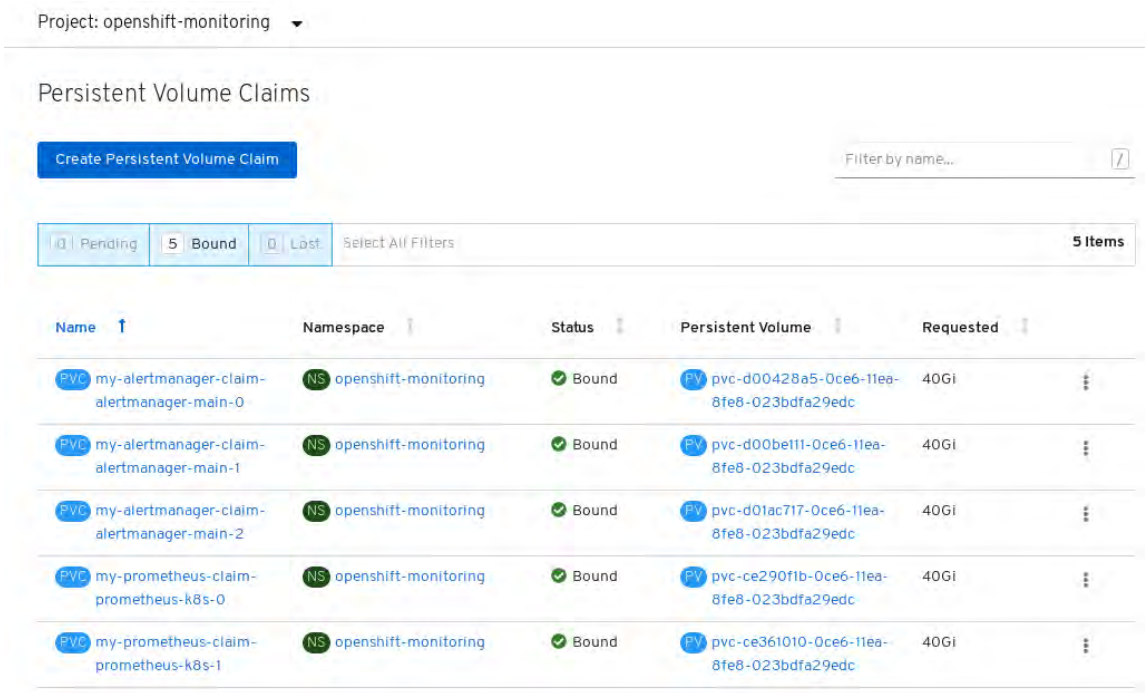
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cluster-monitoring-config
  namespace: openshift-monitoring
data:
  config.yaml: |
      prometheusK8s:
        retention: <time to retain monitoring files, e.g. 24h>
        volumeClaimTemplate:
          metadata:
            name: ocs-prometheus-claim
          spec:
            storageClassName: ocs-storagecluster-ceph-rbd
            resources:
              requests:
                storage: <size of claim, e.g. 40Gi>
      alertmanagerMain:
        volumeClaimTemplate:
          metadata:
            name: ocs-alertmanager-claim
          spec:
            storageClassName: ocs-storagecluster-ceph-rbd
            resources:
              requests:
                storage: <size of claim, e.g. 40Gi>
```

4. Click Create to save and create the Config Map.

What to do next

1. Verify that the Persistent Volume Claims are bound to the pods.
   a. Go to Storage > Persistent Volume Claims.
   b. Set the Project dropdown to openshift-monitoring.
   c. Verify that five Persistent Volume Claims are visible with a state of *Bound*, attached to three `alertmanager-main-*` pods, and two `prometheus-k8s-*` pods. For example, see Figure 1.

Figure 1. Monitoring storage created and bound



2. Verify that the new **`alertmanager-main-*`** pods appear with a state of *Running*.
    a. Go to Workloads > Pods.
    b. Click the new **`alertmanager-main-*`** pods to view the pod details.
    c. From Volumes verify that the volume has a Type, **`ocs-alertmanager-claim`** that matches your new Persistent Volume Claims. For example, **`ocs-alertmanager-claim-alertmanager-main-0`**. For example, see Figure 2.

Figure 2. Persistent Volume Claims attached to **`alertmanager-main-*`** pod



3. Verify that the new **`prometheus-k8s-*`** pods appear with a state of *Running*.
    a. Click the new **`prometheus-k8s-*`** pods to view the pod details.
    b. From Volumes verify that the volume has a Type, **`ocs-prometheus-claim`** that matches your new Persistent Volume Claims. For example, **`ocs-prometheus-claim-prometheus-k8s-0`**. For example, see Figure 3.

Figure 3. Persistent Volume Claims attached to **`prometheus-k8s-*`** pod



Overprovision level policy control [Technology Preview]

Overprovision control is a mechanism that enables you to define a quota on the amount of Persistent Volume Claims (PVCs) consumed from a storage cluster, based on the specific application namespace.

About this task

Important: Technology Preview features are not supported with IBM production service level agreements (SLAs), might not be functionally complete, and IBM does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.
When you enable the overprovision control mechanism, it prevents you from overprovisioning the PVCs consumed from the storage cluster. OpenShift provides flexibility for defining constraints that limit the aggregated resource consumption at cluster scope with the help of **`ClusterResourceQuota`**. For more information see, OpenShift ClusterResourceQuota in *Red Hat OpenShift Container Platform* documentation.

With overprovision control, a **`ClusteResourceQuota`** is initiated, and you can set the storage capacity limit for each storage class. The alarm triggers when 80% of the capacity limit is consumed.

Before you begin

Ensure that the Fusion Data Foundation cluster is created.

Procedure

1. Deploy `storagecluster` either from the command line interface or the user interface.
2. Label the application namespace.

```
apiVersion: v1
kind: Namespace
metadata:
     name: <desired_name>
     labels:
          storagequota: <desired_label>
```

<desired_name>
: Specify a name for the application namespace, for example, `quota-rbd`.

<desired_label>
: Specify a label for the storage quota, for example, `storagequota1`.

3. Edit the `storagecluster` to set the quota limit on the storage class, where *<ocs_storagecluster_name>* specifies the name of the storage cluster.

```
oc edit storagecluster -n openshift-storage <ocs_storagecluster_name>
```

4. Add an entry for Overprovision Control with the desired hard limit into the StorageCluster.Spec.

```
apiVersion: ocs.openshift.io/v1
kind: StorageCluster
spec:
 [...]
     overprovisionControl:
     - capacity: <desired_quota_limit>
          storageClassName: <storage_class_name>
          quotaName: <desired_quota_name>
          selector:
            labels:
                  matchLabels:
                   storagequota: <desired_label>
[...]
```

<desired_quota_limit>
: Specify a desired quota limit for the storage class, for example, `27Ti`.

<storage_class_name>
: Specify the name of the storage class for which you want to set the quota limit, for example, `ocs-storagecluster-ceph-rbd`.

<desired_quota_name>
: Specify a name for the storage quota, for example, `quota1`.

<desired_label>
: Specify a label for the storage quota, for example, `storagequota1`.

5. Save the modified `storagecluster`.
6. Verify that the `clusterresourcequota` is defined.
   Note: `clusterresourcequota` should reflect the `quotaName` that you defined in the previous step, for example, `quota1`.

```
oc get clusterresourcequota -A
```

```
oc describe clusterresourcequota -A
```

Cluster logging for Fusion Data Foundation

You can deploy cluster logging to aggregate logs for a range of OpenShift Container Platform services.

For information about how to deploy cluster logging, see Observability > Logging in Red Hat OpenShift Container Platform product documentation.

Upon initial OpenShift Container Platform deployment, Fusion Data Foundation is not configured by default and the OpenShift Container Platform cluster relies solely on default storage available from the nodes. You can edit the default configuration of OpenShift logging (ElasticSearch) to be backed by Fusion Data Foundation to have Fusion Data Foundation backed logging (Elasticsearch).

Important: Always ensure that you have plenty of storage capacity for these services. If you run out of storage space for these critical services, the logging application becomes inoperable and very difficult to recover.
IBM recommends configuring shorter curation and retention intervals for these services. For more information, see Observability > Logging in Red Hat OpenShift Container Platform product documentation.

If you run out of storage space for these services, contact IBM Support.

- **Configuring persistent storage**
  You can configure a persistent storage class and size for the Elasticsearch cluster using the storage class name and size parameters. The Cluster Logging Operator creates a Persistent Volume Claim for each data node in the Elasticsearch cluster based on these parameters.
- **Configuring cluster logging to use Fusion Data Foundation**
  Configure Fusion Data Foundation as storage for the Fusion Data Foundation cluster logging.

Configuring persistent storage

You can configure a persistent storage class and size for the Elasticsearch cluster using the storage class name and size parameters. The Cluster Logging Operator creates a Persistent Volume Claim for each data node in the Elasticsearch cluster based on these parameters.

The following example specifies that each data node in the cluster is bound to a Persistent Volume Claim that requests `200GiB` of `ocs-storagecluster-ceph-rbd` storage. Each primary shard is backed by a single replica. A copy of the shard is replicated across all the nodes and are always available and the copy can be recovered if at least two nodes exist due to the single redundancy policy. For information about Elasticsearch replication policies, see Observability > Logging in Red Hat OpenShift Container Platform product documentation.

```
spec:
    logStore:
      type: "elasticsearch"
      elasticsearch:
        nodeCount: 3
        storage:
          storageClassName: "ocs-storagecluster-ceph-rbd"
          size: "200G"
```

Note: Omission of the storage block results in a deployment backed by default storage. For example:

```
spec:
    logStore:
      type: "elasticsearch"
      elasticsearch:
        nodeCount: 3
        storage: {}
```

Configuring cluster logging to use Fusion Data Foundation

Configure Fusion Data Foundation as storage for the Fusion Data Foundation cluster logging.

About this task

Note: You can obtain all the logs when you configure logging for the first time in Fusion Data Foundation. However, after you uninstall and reinstall logging, the old logs are removed and only the new logs are processed.

Before you begin

Ensure you have the following:

- You have administrative access to OpenShift Web Console.
- Fusion Data Foundation Operator is installed and running in the `openshift-storage` namespace.
- Cluster logging Operator is installed and running in the `openshift-logging` namespace.

Procedure

1. Go to Administration > Custom Resource Definitions.
2. From the Custom Resource Definitions page, click ClusterLogging.
3. From the Custom Resource Definition Overview page, select View Instances. from the Actions menu.
   Alternatively, click the Instances tab.
4. From the Cluster Logging page, click Create Cluster Logging.
   A refresh might be necessary to load the data.
5. In the YAML, replace the storageClassName with the `storageclass` that uses the provisioner `openshift-storage.rbd.csi.ceph.com`.
   In the following example, the name of the storageclass is `ocs-storagecluster-ceph-rbd`.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: ocs-storagecluster-ceph-rbd
        size: 200G  # Change as per your requirement
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana"
    kibana:
      replicas: 1
  curation:
    type: "curator"
    curator:
      schedule: "30 3 * * *"
  collection:
    logs:
```

```
      type: "fluentd"
      fluentd: {}
```

If you have tainted the Fusion Data Foundation nodes, you must add toleration to enable scheduling of the daemonset pods for logging.

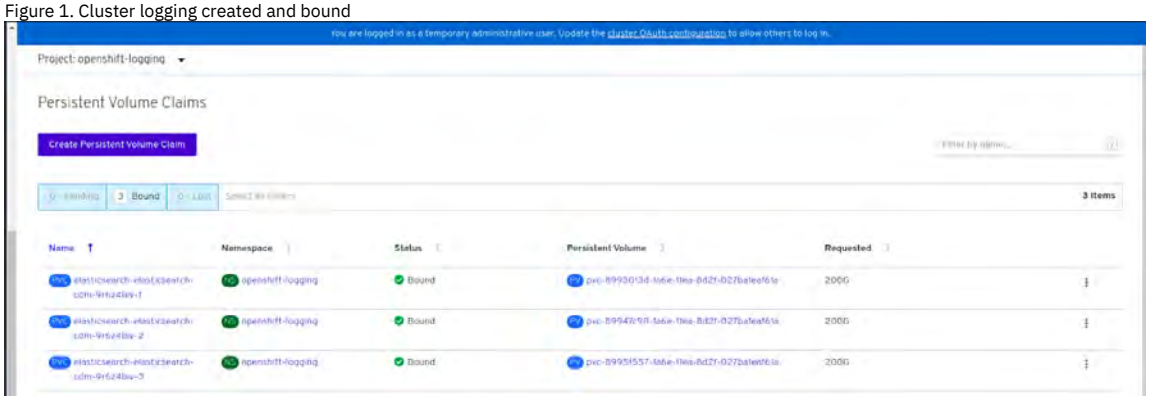
```
spec:
[...]
  collection:
    logs:
      fluentd:
        tolerations:
        - effect: NoSchedule
          key: node.ocs.openshift.io/storage
          value: 'true'
      type: fluentd
```

6. Click Save.

What to do next

1. Verify that the Persistent Volume Claims are bound to the `elasticsearch` pods.
   a. Go to Storage > Persistent Volume Claims.
   b. Set the Project dropdown to openshift-logging.
   c. Verify that Persistent Volume Claims are visible with a state of *Bound*, attached to `elasticsearch-*` pods.

   Figure 1. Cluster logging created and bound



2. Verify that the new cluster logging is being used.
   a. Click Workload > Pods.
   b. Set the Project to `openshift-logging`.
   c. Verify that the new `elasticsearch-*` pods appear with a state of *Running*.
   d. Click the new `elasticsearch-*` pod to view pod details.
   e. From Volumes verify that the `elasticsearch` volume has a Type that matches your new Persistent Volume Claim. For example, `elasticsearch-elasticsearch-cdm-9r624biv-3`.
   f. Click the Persistent Volume Claim name and verify the storage class name in the PersistentVolumeClaim Overview page.

Note:

- Ensure to use a shorter curator time to avoid PV full scenario on PVs attached to Elasticsearch pods.
  You can configure Curator to delete Elasticsearch data based on retention settings. It is recommended that you set the following default index data retention of 5 days as a default.

```
config.yaml: |
    openshift-storage:
      delete:
        days: 5
```

  For more information about Elasticsearch, see Observability > Logging in Red Hat OpenShift Container Platform product documentation.
- To uninstall the cluster logging backed by Persistent Volume Claim, use the procedure removing the cluster logging operator from Fusion Data Foundation in the uninstall chapter of the respective deployment guide.

Creating Multus networks

Multus networks enable pods to attach additional network interfaces by creating `NetworkAttachmentDefinition` custom resources; Fusion Data Foundation uses the macvlan CNI plug-in to provide each pod with a unique MAC address on the additional network.

Red Hat OpenShift Container Platform uses the Multus CNI plug-in to allow chaining of CNI plug-ins. During cluster installation, you can configure your default pod network. The default network handles all ordinary network traffic for the cluster.

You can define an extra network based on the available CNI plug-ins and attach one or more of these networks to your pods. To attach more network interfaces to a pod, you must create configurations that define how the interfaces are attached.

You specify each interface by using a NetworkAttachmentDefinition custom resource (CR). A CNI configuration inside each of the NetworkAttachmentDefinition defines how that interface is created.

Fusion Data Foundation uses the CNI plug-in called macvlan. Creating a macvlan-based additional network allows pods on a host to communicate with other hosts and pods on those hosts by using a physical network interface. Each pod that is attached to a macvlan-based additional network is provided a unique MAC address.

- **Creating network attachment definitions**
  The Multus networks that you create depend on the number of available network interfaces you have for Fusion Data Foundation traffic. You can separate all of the storage traffic onto one of two interfaces where one interface that is used for default Red Hat OpenShift SDN or to further separate storage traffic into client storage traffic (public) and storage replication traffic (private or cluster).

Creating network attachment definitions

The Multus networks that you create depend on the number of available network interfaces you have for Fusion Data Foundation traffic. You can separate all of the storage traffic onto one of two interfaces where one interface that is used for default Red Hat OpenShift SDN or to further separate storage traffic into client storage traffic (public) and storage replication traffic (private or cluster).

To use Multus, an already working cluster with the correct networking configuration is required. For more information, see Requirements for Multus configuration. The newly created **NetworkAttachmentDefinition** (NAD) can be selected during the Storage Cluster installation and must be created before the Storage Cluster.

Note: Network attachment definitions can only use the **whereabouts** IP address management (IPAM), and it must specify the **range** field. **ipRanges** and plugin chaining are not supported.
The following example of **NetworkAttachmentDefinition** for all storage traffic, public, and cluster, on the same interface. It requires one additional interface on all schedulable nodes (Red Hat OpenShift default SDN on separate network interface).

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-public-cluster
  namespace: openshift-storage
spec:
  config: '{
        "cniVersion": "0.3.1",
        "type": "macvlan",
        "master": "ens2",
        "mode": "bridge",
        "ipam": {
            "type": "whereabouts",
            "range": "192.168.1.0/24"
        }
  }'
```

Note:
All network interface names must be the same on all the nodes that are attached to the Multus network (that is, **ens2** for **ocs-public-cluster**).

The following is an example **NetworkAttachmentDefinition** for storage traffic on separate Multus networks, public, for client storage traffic, and cluster, for replication traffic. It requires two additional interfaces on Red Hat OpenShift nodes hosting OSD pods and one additional interface on all other schedulable nodes (Red Hat OpenShift default SDN on separate network interface).

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-public
  namespace: openshift-storage
spec:
  config: '{
        "cniVersion": "0.3.1",
        "type": "macvlan",
        "master": "ens2",
        "mode": "bridge",
        "ipam": {
            "type": "whereabouts",
            "range": "192.168.1.0/24"
        }
  }'
```

Example of **NetworkAttachmentDefinition**:

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-cluster
  namespace: openshift-storage
spec:
  config: '{
        "cniVersion": "0.3.1",
        "type": "macvlan",
        "master": "ens3",
        "mode": "bridge",
        "ipam": {
            "type": "whereabouts",
            "range": "192.168.2.0/24"
        }
  }'
```

Note:
All network interface names must be the same on all the nodes that are attached to the Multus networks (that is, **ens2** for **ocs-public**, and **ens3** for **ocs-cluster**).

Backing OpenShift Container Platform applications with Fusion Data Foundation

Configure OpenShift Container Platform applications to use Fusion Data Foundation.

About this task

You cannot directly install Fusion Data Foundation during the OpenShift Container Platform installation. However, you can install Fusion Data Foundation on an existing OpenShift Container Platform by using the Operator Hub and then configure the OpenShift Container Platform applications to be backed by Fusion Data Foundation.

Before you begin

Ensure the following:

- OpenShift Container Platform is installed and you have administrative access to OpenShift Web Console.
- Fusion Data Foundation is installed and running in the `openshift-storage` namespace.

Procedure

1. From the OpenShift Web Console, create a new deployment in one of the following ways.
   - Go to Workloads > Deployments.
     From the Deployments page, you can do one of the following:
     - Select any existing deployment and click Action menu > Add Storage.
     - Create a new deployment and then add storage.
       - Click Create Deployment to create a new deployment.
       - Edit the YAML based on your requirement to create a deployment.
       - Click Create.
       - Select Action menu > Add Storage.
   - Go to Workloads > Deployment Configs.
     From the Deployment Configs page, you can do one of the following:
     - Select any existing deployment and click Action menu > Add Storage.
     - Create a new deployment and then add storage.
       - Click Create Deployment Config to create a new deployment.
       - Edit the YAML based on your requirement to create a deployment.
       - Click Create.
       - Select Action menu > Add Storage.
2. In the Add Storage page, you can choose one of the following options:
   - Click the Use existing claim option and select a suitable PVC from the drop-down list.
   - Click the Create new claim option.
     - Select the appropriate `CephFS` or `RBD` storage class from the Storage Class drop-down list.
     - Provide a name for the Persistent Volume Claim.
     - Select ReadWriteAfter (RWO) or ReadWriteMany (RWX) access mode.
       Note: ReadOnlyMany (ROX) is deactivated as it is not supported.
     - Select the size of the desired storage capacity.
       Note: You can expand the block PVs but cannot reduce the storage capacity after the creation of Persistent Volume Claim.
3. Optional: If required, specify the mount path and subpath for the mount path volume inside the container.
4. Click Save.

What to do next

1. Depending on your configuration, perform one of the following:
   - Go to Workloads > Deployments.
   - Click Workloads > Deployment Configs.
2. Set the Project as required.
3. Click the deployment for which you added storage to display the deployment details.
4. From Volumes verify that your deployment has a Type that matches the assigned Persistent Volume Claim.
5. Click the Persistent Volume Claim name and verify the storage class name in the Persistent Volume Claim Overview page.

Adding file and object storage to an existing external Fusion Data Foundation cluster

Add file storage (using Metadata Servers) or object storage (using Ceph Object Gateway) or both to an external Fusion Data Foundation cluster that was initially deployed to provide only block storage.

About this task

When Fusion Data Foundation is configured in external mode, there are several ways to provide storage for persistent volume claims and object bucket claims.

- Persistent volume claims for block storage are provided directly from the external IBM Storage Ceph cluster.
- Persistent volume claims for file storage can be provided by adding a Metadata Server (MDS) to the external IBM Storage Ceph cluster.
- Object bucket claims for object storage can be provided either by using the Multicloud Object Gateway or by adding the Ceph Object Gateway to the external IBM Storage Ceph cluster.

Before you begin

Ensure you have the following:

- Fusion Data Foundation is installed and running on the corresponding OpenShift Container Platform version with the Fusion Data Foundation cluster in external mode is in the *Ready* state.
- Your external IBM Storage Ceph cluster is configured with one or both of the following:
  - A Ceph Object Gateway (RGW) endpoint that can be accessed by the OpenShift Container Platform cluster for object storage.
  - A Metadata Server (MDS) pool for file storage.
- Ensure that you know the parameters used with the ceph-external-cluster-details-exporter.py script during external Fusion Data Foundation cluster deployment.

Procedure

1. Download the Fusion Data Foundation version of the ceph-external-cluster-details-exporter.py Python script by using CSV or ConfigMap.
   Important: Starting with Fusion Data Foundation version 4.19, downloading the `ceph-external-cluster-details-exporter.py` Python script using a CSV is no longer supported. The only supported method is using a ConfigMap.
   Command examples are as follows:
   - **CSV**

     ```
     oc get csv $(oc get csv -n openshift-storage | grep rook-ceph-operator | awk '{print $1}') -n openshift-storage -o
     jsonpath='{.metadata.annotations.externalClusterScript}' | base64 --decode >ceph-external-cluster-details-exporter.py
     ```

   - **ConfigMap**

     ```
     oc get cm rook-ceph-external-cluster-script-config -n openshift-storage -o jsonpath='{.data.script}' | base64 --
     decode > ceph-external-cluster-details-exporter.py
     ```

2. Update permission caps on the external IBM Storage Ceph cluster by running `ceph-external-cluster-details-exporter.py` on any client node in the external IBM Storage Ceph cluster.
   Your IBM Storage Ceph administrator need to run this command.

   ```
   python3 ceph-external-cluster-details-exporter.py --upgrade \
   --run-as-user=ocs-client-name \
   --rgw-pool-prefix rgw-pool-prefix
   ```

   `--run-as-user`
   : The client name used during Fusion Data Foundation cluster deployment. Use the default client name `client.healthchecker` if a different client name was not set.

   `--rgw-pool-prefix`
   : The prefix used for the Ceph Object Gateway pool. This can be omitted if the default prefix is used.

3. Generate and save configuration details from the external IBM Storage Ceph cluster.
   a. Generate configuration details by running ceph-external-cluster-details-exporter.py on any client node in the external IBM Storage Ceph cluster.

      ```
      # python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name rbd-block-pool-name --monitoring-endpoint
      ceph-mgr-prometheus-exporter-endpoint --monitoring-endpoint-port ceph-mgr-prometheus-exporter-port --run-as-user ocs-
      client-name  --rgw-endpoint rgw-endpoint --rgw-pool-prefix rgw-pool-prefix
      ```

      `--monitoring-endpoint`
      : (Optional) It accepts comma separated list of IP addresses of active and standby mgrs reachable from the OpenShift Container Platform cluster. If not provided, the value is automatically populated.

      `--monitoring-endpoint-port`
      : (Optional) It is the port associated with the ceph-mgr Prometheus exporter specified by `--monitoring-endpoint`. If not provided, the value is automatically populated.

      `--run-as-user`
      : The client name used during Fusion Data Foundation cluster deployment. Use the default client name client.healthchecker if a different client name was not set.

      `--rgw-endpoint`
      : (Optional) Provide this parameter to provision object storage through Ceph Object Gateway for Fusion Data Foundation.

      `--rgw-pool-prefix`
      : The prefix used for the Ceph Object Gateway pool. This can be omitted if the default prefix is used.

      User permissions are updated as shown:

      ```
      caps: [mgr] allow command config
      caps: [mon] allow r, allow command quorum_status, allow command version
      caps: [osd] allow rwx pool=default.rgw.meta, allow r pool=.rgw.root, allow rw pool=default.rgw.control, allow rx
      pool=default.rgw.log, allow x pool=default.rgw.buckets.index
      ```

      Note: Ensure that all the parameters (including the optional arguments) except the Ceph Object Gateway details (if provided), are the same as what was used during the deployment of Fusion Data Foundation in external mode.
   b. Save the output of the script in an external-cluster-config.json file.
      The following example output shows the generated configuration changes in **bold text**.

      ```
      [{"name": "rook-ceph-mon-endpoints", "kind": "ConfigMap", "data": {"data": "xxx.xxx.xxx.xxx:xxxx", "maxMonId": "0",
      "mapping": "{}"}}, {"name": "rook-ceph-mon", "kind": "Secret", "data": {"admin-secret": "admin-secret", "fsid": "<fs-
      id>", "mon-secret": "mon-secret"}}, {"name": "rook-ceph-operator-creds", "kind": "Secret", "data": {"userID": "<user-
      id>", "userKey": "<user-key>"}}, {"name": "rook-csi-rbd-node", "kind": "Secret", "data": {"userID": "csi-rbd-node",
      "userKey": "<user-key>"}}, {"name": "ceph-rbd", "kind": "StorageClass", "data": {"pool": "<pool>"}}, {"name":
      "monitoring-endpoint", "kind": "CephCluster", "data": {"MonitoringEndpoint": "xxx.xxx.xxx.xxx", "MonitoringPort":
      "xxxx"}}, {"name": "rook-ceph-dashboard-link", "kind": "Secret", "data": {"userID": "ceph-dashboard-link", "userKey":
      "<user-key>"}}, {"name": "rook-csi-rbd-provisioner", "kind": "Secret", "data": {"userID": "csi-rbd-provisioner",
      "userKey": "<user-key>"}}, {"name": "rook-csi-cephfs-provisioner", "kind": "Secret", "data": {"adminID": "csi-cephfs-
      provisioner", "adminKey": "<admin-key>"}}, {"name": "rook-csi-cephfs-node", "kind": "Secret", "data": {"adminID":
      "csi-cephfs-node", "adminKey": "<admin-key>"}}, {"name": "cephfs", "kind": "StorageClass", "data": {"fsName":
      "cephfs", "pool": "cephfs_data"}}, {"name": "ceph-rgw", "kind": "StorageClass", "data": {"endpoint":
      "xxx.xxx.xxx.xxx:xxxx", "poolPrefix": "default"}}, {"name": "rgw-admin-ops-user", "kind": "Secret", "data":
      {"accessKey": "<access-key>", "secretKey": "<secret-key>"}}]
      ```

4. Upload the generated JSON file.
   a. Log in to the OpenShift web console.
   b. Click Workloads > Secrets.

c. Set Project to **openshift-storage**.
d. Click **rook-ceph-external-cluster-details**.
e. Go to Actions > Edit Secret.
f. Click Browse and upload the external-cluster-config.json file.
g. Click Save.

What to do next

- To verify that the Fusion Data Foundation cluster is healthy and data is resilient, go to Storage > Data Foundation > Storage Systems tab and then click on the storage system name.
  - From the Overview > Block and File tab, check the Status card to confirm that the Storage Cluster has a green tick indicating it is healthy.
- If you added a Metadata Server for file storage:
  1. Go to Workloads > Pods and verify that **csi-cephfsplugin-*** pods are created new and are in the *Running* state.
  2. Go to Storage > Storage Classes and verify that the **ocs-external-storagecluster-cephfs** storage class is created.
- If you added the Ceph Object Gateway for object storage:

  1. Go to Storage > Storage Classes and verify that the **ocs-external-storagecluster-ceph-rgw** storage class is created.
  2. To verify that the Fusion Data Foundation cluster is healthy and data is resilient, go to Storage > Data Foundation > Storage Systems tab and then click on the storage system name.
  3. From the Object tab, confirm Object Service and Data resiliency has a green tick indicating it is healthy.

How to use dedicated worker nodes for Fusion Data Foundation

Use dedicated worker nodes for IBM Fusion Data Foundation

Any Red Hat OpenShift Container Platform subscription requires an Fusion Data Foundation subscription. However, you can save on the OpenShift Container Platform subscription costs if you are using infrastructure nodes to schedule Fusion Data Foundation resources.

When storage is added into OpenShift Container Platform cluster, note the following:

- There is no need to remove **worker** node label from storage nodes
- Configure the Data Foundation storage cluster and enable the **Dedicated nodes for infrastructure** check box. IBM Fusion automatically converts the storage nodes to infrastructure nodes.

- **Anatomy of an Infrastructure node**
  Understand the infrastructure node attributes.

Anatomy of an Infrastructure node

Understand the infrastructure node attributes.

Infrastructure nodes for use with Fusion Data Foundation have a few attributes. The **infra** node-role label is required to ensure the node does not consume Red Hat OpenShift Container Platform entitlements. The **infra** node-role label is responsible for ensuring only Fusion Data Foundation entitlements are necessary for the nodes running Fusion Data Foundation.

- Labeled with **node-role.kubernetes.io/infra**

Adding an Fusion Data Foundation taint with a **NoSchedule** effect is also required so that the **infra** node only schedules Fusion Data Foundation resources.

- Tainted with **node.ocs.openshift.io/storage="true"**

The label identifies the Red Hat OpenShift Container Platform node as an **infra** node so that Red Hat OpenShift Container Platform subscription cost is not applied. The taint prevents non Fusion Data Foundation resources to be scheduled on the tainted nodes.
Note: Adding storage taint on nodes might require toleration handling for the other **daemonset** pods such as **openshift-dns daemonset**. For information about how to manage the tolerations, see [RHOCP 4.x] openshift-dns daemonset doesn't include toleration to run on nodes with taints on the Red Hat Customer Portal.
Example of the taint and labels required on infrastructure node that is used to run Fusion Data Foundation services:

```
kind: Node
apiVersion: v1
metadata:
  creationTimestamp: null
  labels:
    node-role.kubernetes.io/worker: ""
    node-role.kubernetes.io/infra: ""
    cluster.ocs.openshift.io/openshift-storage: ""
  # ...
spec:
  taints:
  - effect: NoSchedule
    key: node.ocs.openshift.io/storage
    value: "true"
  # ...
status:
  # ...
```

Managing persistent volume claims

Manage and automate the fulfillment of Persistent Volume Claim requests.

- **Configuring application pods to use Fusion Data Foundation**
  Follow the instructions in this section to configure Fusion Data Foundation as storage for an application pod.
- **Viewing Persistent Volume Claim request status**
  Use this procedure to view the status of a PVC request.
- **Reviewing Persistent Volume Claim request events**
  Use this procedure to review and address Persistent Volume Claim (PVC) request events.
- **Expanding Persistent Volume Claims**
  Fusion Data Foundation has the ability to expand Persistent Volume Claims providing more flexibility in the management of persistent storage resources.
- **Dynamic provisioning**
  The StorageClass resource object describes and classifies storage that can be requested, as well as provides a means for passing parameters for dynamically provisioned storage on demand. StorageClass objects can also serve as a management mechanism for controlling different levels of storage and access to the storage. Cluster Administrators (`cluster-admin`) or Storage Administrators (`storage-admin`) define and create the StorageClass objects that users can request without needing any intimate knowledge about the underlying storage volume sources.

Configuring application pods to use Fusion Data Foundation

Follow the instructions in this section to configure Fusion Data Foundation as storage for an application pod.

Before you begin

Ensure you have the following

- Administrative access to OpenShift Web Console.
- Fusion Data Foundation Operator is installed and running in the `openshift-storage` namespace. From OpenShift Web Console, go to Operators > Installed Operators to view installed operators.
- The default storage classes provided by Fusion Data Foundation are available. From OpenShift Web Console, go to Storage > StorageClasses to view default storage classes.

Procedure

1. Create a Persistent Volume Claim (PVC) for the application to use.
   a. From OpenShift Web Console, go to Storage > Persistent Volume Claims.
   b. Set the Project for the application pod.
   c. Click Create Persistent Volume Claim.
      i. Specify a Storage Class provided by Fusion Data Foundation.
      ii. Specify the PVC Name, for example, `myclaim`.
      iii. Select the required Access Mode.
      Note: The `Shared access (RWX)` Access Mode is not supported by IBM FlashSystem storage systems.
      iv. For RADOS Block Device (RBD), if the Access Mode is ReadWriteOnce (`RWO`), select the required Volume mode. The default volume mode is `Filesystem`.
      v. Specify a Size as per application requirement.
      vi. Click Create and wait until the PVC is in *Bound* status.
2. Configure a new or existing application pod to use the new PVC.

   For a new application pod, perform the following steps:
   a. Go to Workloads > Pods.
   b. Create a new application pod.
   c. Under the `spec:` section, add `volumes:` section to add the new PVC as a volume for the application pod.

```
volumes:
  - name: <volume_name>
    persistentVolumeClaim:
      claimName: <pvc_name>
```

   For example:

```
volumes:
  - name: mypd
    persistentVolumeClaim:
      claimName: myclaim
```

   For an existing application pod, perform the following steps:
   a. Go to Workloads > Deployment Configs.
   b. Search for the required deployment config associated with the application pod and click on its Action Menu > Edit Deployment Config
   c. Under the `spec:` section, add `volumes:` section to add the new PVC as a volume for the application pod.

```
volumes:
  - name: <volume_name>
    persistentVolumeClaim:
      claimName: <pvc_name>
```

   For example:

```
volumes:
  - name: mypd
    persistentVolumeClaim:
      claimName: myclaim
```

d. Click Save.

3. Verify that the new configuration is being used.
   a. Go to Workloads > Pods.
   b. Set the Project for the application pod.
   c. Verify that the application pod appears in a *Running* state.
   d. Click the application pod name to view pod details.
   e. From Volumes verify that the volume has a Type that matches your new Persistent Volume Claim. For example, `myclaim`.

Viewing Persistent Volume Claim request status

Use this procedure to view the status of a PVC request.

Before you begin

Be sure you have Administrator access to Fusion Data Foundation.

Procedure

1. Log in to OpenShift Web Console and go to Storage > Persistent Volume Claims.
2. Search for the required PVC name by using the Filter textbox.
   You can also filter the list of PVCs by Name or Label to narrow down the list.
3. Check the Status column corresponding to the required PVC.
4. Click the required Name to view the PVC details.

Reviewing Persistent Volume Claim request events

Use this procedure to review and address Persistent Volume Claim (PVC) request events.

Before you begin

Be sure you have Administrator access to OpenShift Web Console.

Procedure

1. From the OpenShift Web Console, go to Storage > Data Foundation.
2. From the Storage systems tab, select the storage system and then click Overview > Block and File.
3. Locate the Inventory card to see the number of PVCs with errors.
4. Go to Storage > Persistent Volume Claims.
5. Search for the required PVC name by using the Filter textbox.
6. Check the PVC name and navigate to Events.
7. Address the events as required or as directed.

Expanding Persistent Volume Claims

Fusion Data Foundation has the ability to expand Persistent Volume Claims providing more flexibility in the management of persistent storage resources.

About this task

Expansion is supported for the following Persistent Volumes:

- PVC with ReadWriteAfter (RWO) and ReadWriteMany (RWX) access that is based on Ceph File System (CephFS) for volume mode `Filesystem`.
- PVC with ReadWriteAfter (RWO) access that is based on Ceph RADOS Block Devices (RBDs) with volume mode `Filesystem`.
- PVC with ReadWriteAfter (RWO) access that is based on Ceph RADOS Block Devices (RBDs) with volume mode `Block`.
- PVC with ReadWriteOncePod (RWOP) that is based on Ceph File System (CephFS) or Network File System (NFS) for volume mode `Filesystem`.
- PVC with ReadWriteOncePod (RWOP) access that is based on Ceph RADOS Block Devices (RBDs) with volume mode `Filesystem`. With RWOP access mode, you mount the volume as read-write by a single pod on a single node.

Note: PVC expansion is not supported for OSD, MON and encrypted PVCs.

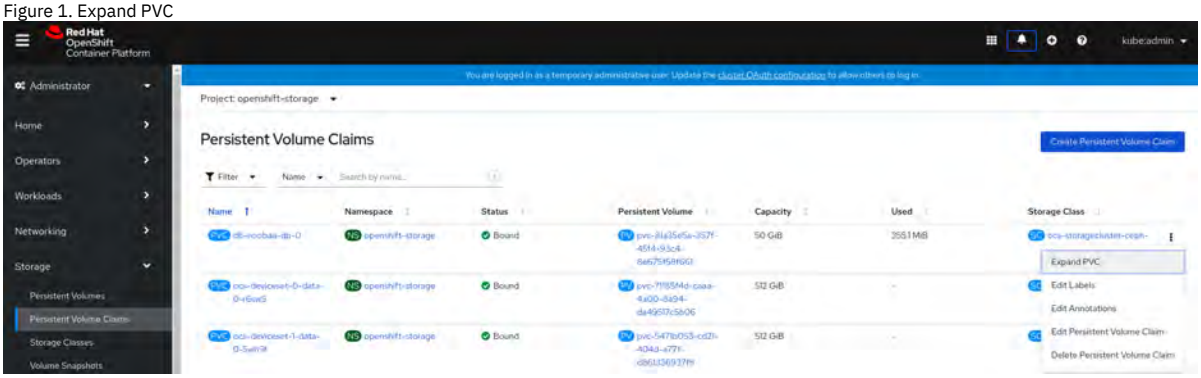
Before you begin

Be sure you have Administrator access to OpenShift Web Console.

Procedure

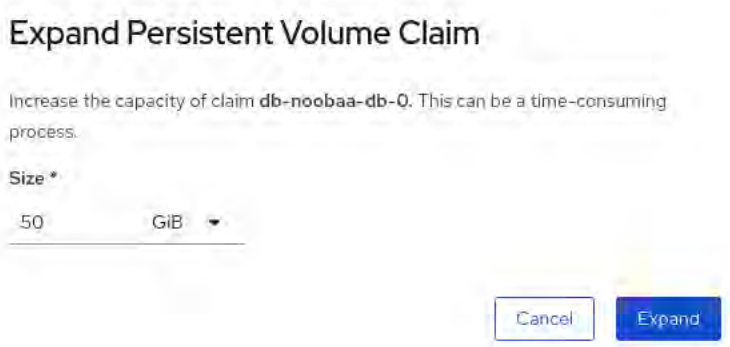
1. From the OpenShift Web Console, go to Storage > Persistent Volume Cliams.
2. From the Persistent Volume Claim that you want to expand, go to Action Menu > Expand PVC.

Figure 1. Expand PVC



3. Select the new size of the Persistent Volume Claim, then click Expand.

Figure 2. Expand Persistent Volume Claim form



4. To verify the expansion, navigate to the PVC's details page and verify the Capacity field has the correct size requested.
   Note: When expanding PVCs based on Ceph RADOS Block Devices (RBDs), if the PVC is not already attached to a pod the `Condition type` is `FileSystemResizePending` in the PVC's details page. After the volume is mounted, filesystem resize succeeds and the new size is reflected in the Capacity field.

Dynamic provisioning

The StorageClass resource object describes and classifies storage that can be requested, as well as provides a means for passing parameters for dynamically provisioned storage on demand. StorageClass objects can also serve as a management mechanism for controlling different levels of storage and access to the storage. Cluster Administrators (`cluster-admin`) or Storage Administrators (`storage-admin`) define and create the StorageClass objects that users can request without needing any intimate knowledge about the underlying storage volume sources.

The Fusion Data Foundation persistent volume framework enables this functionality and allows administrators to provision a cluster with persistent storage. The framework also gives users a way to request those resources without having any knowledge of the underlying infrastructure.

Many storage types are available for use as persistent volumes in Fusion Data Foundation. While all of them can be statically provisioned by an administrator, some types of storage are created dynamically using the built-in provider and plug-in APIs.

- **Dynamic provisioning in Fusion Data Foundation**
  IBM Fusion Data Foundation is software-defined storage that is optimized for container environments. It runs as an operator on OpenShift Container Platform to provide highly integrated and simplified persistent storage management for containers.
- **Available dynamic provisioning plug-ins**
  Fusion Data Foundation provides provisioner plugins, which have generic implementations for dynamic provisioning that use the cluster's configured provider's API to create new storage resources.

Dynamic provisioning in Fusion Data Foundation

IBM Fusion Data Foundation is software-defined storage that is optimized for container environments. It runs as an operator on OpenShift Container Platform to provide highly integrated and simplified persistent storage management for containers.

Fusion Data Foundation supports a variety of storage types, including:

- Block storage for databases.
- Shared file storage for continuous integration, messaging, and data aggregation.
- Object storage for archival, backup, and media storage.

Fusion Data Foundation uses IBM Storage Ceph to provide the file, block, and object storage that backs persistent volumes, and Rook.io to manage and orchestrate provisioning of persistent volumes and claims. NooBaa provides object storage, and its Multicloud Gateway allows object federation across multiple cloud environments (available as a Technology Preview).

In Fusion Data Foundation the IBM Storage Ceph Container Storage Interface (CSI) driver for RADOS Block Device (RBD) and Ceph File System (CephFS) handles the dynamic provisioning requests. When a PVC request comes in dynamically, the CSI driver has the following options:

- Create a PVC with ReadWriteAfter (RWO) and ReadWriteMany (RWX) access that is based on Ceph RBDs with volume mode `Block`
- Create a PVC with ReadWriteAfter (RWO) access that is based on Ceph RBDs with volume mode `Filesystem`
- Create a PVC with ReadWriteAfter (RWO) and ReadWriteMany (RWX) access that is based on CephFS for volume mode `Filesystem`
- Create a PVC with ReadWriteOncePod (RWOP) access that is based on CephFS, NFS, and RBD. With RWOP access mode, you mount the volume as read-write by a single pod on a single node.

The judgment of which driver (RBD or CephFS) to use is based on the entry in the storageclass.yaml file.

Available dynamic provisioning plug-ins

Fusion Data Foundation provides provisioner plugins, which have generic implementations for dynamic provisioning that use the cluster's configured provider's API to create new storage resources.

Table 1 details the provided Fusion Data Foundation provisioner plugins.

Table 1. Available dynamic provisioner plugins

Storage type	Provisioner plug-in name	Notes
OpenStack Cinder	`kubernetes.io/cinder`	
AWS Elastic Block Store (EBS)	`kubernetes.io/aws-ebs`	For dynamic provisioning when using multiple clusters in different zones, tag each node with `Key=kubernetes.io/cluster/<cluster_name>,Value=<cluster_id>` where `<cluster_name>` and `<cluster_id>` are unique per cluster.
AWS Elastic File System (EFS)		Dynamic provisioning is accomplished through the EFS provisioner pod and not through a provisioner plug-in.
Azure Disk	`kubernetes.io/azure-disk`	
Azure File	`kubernetes.io/azure-file`	The `persistent-volume-binder` ServiceAccount requires permissions to create and get Secrets to store the Azure storage account and keys.
GCE Persistent Disk (gcePD)	`kubernetes.io/gce-pd`	In multi-zone configurations, it is advisable to run one Fusion Data Foundation cluster per GCE project to avoid PVs from being created in zones where no node in the current cluster exists.
VMware vSphere	`kubernetes.io/vsphere-volume`	

Important: Any chosen provisioner plug-in also requires configuration for the relevant cloud, host, or third-party provider as per the relevant documentation.

Reclaiming space on target volumes

Reclaim the actual available storage space on target volumes.

The deleted files or chunks of zero data sometimes take up storage space on the Ceph cluster resulting in inaccurate reporting of the available storage space. The reclaim space operation removes such discrepancies by executing the following operations on the target volume:

- `fstrim` - This operation is executed on volumes that are in `Filesystem` mode and only if the volume is mounted to a pod at the time of execution of reclaim space operation.
- `rbd sparsify` - This operation is executed when the volume is not attached to any pods and reclaims the space occupied by chunks of 4M-sized zeroed data.

Note:

- The reclaim space operation is supported only by the Ceph RBD volumes.
- The reclaim space operation involves a performance penalty when it is being executed.

You can use one of the following methods to reclaim the space:

- Enabling reclaim space operation using Annotating PersistentVolumeClaims (Recommended method to use for enabling reclaim space operation)
- Enabling reclaim space operation using ReclaimSpaceJob
- Enabling reclaim space operation using ReclaimSpaceCronJob

- **Enabling reclaim space operation using Annotating Persistent Volume Claims**
  Enable reclaim space operations using Annotating Persistent Volume Claims.
- **Disabling reclaim space for a specific PersistentVolumeClaim**
  To disable reclaim space for a specific PersistentVolumeClaim (PVC), modify the associated `ReclaimSpaceCronJob` custom resource (CR).
- **Enabling reclaim space operation using ReclaimSpaceJob**
  `ReclaimSpaceJob` is a namespaced custom resource designed to invoke reclaim space operation on the target volume.
- **Enabling reclaim space operation using ReclaimSpaceCronJob**
  `ReclaimSpaceCronJob` invokes the reclaim space operation based on the given schedule (daily, weekly, and so on). You have to create `ReclaimSpaceCronJob` one time only for a persistent volume claim. The `CSI-addons` controller creates a `ReclaimSpaceJob` at the requested time and interval with the schedule attribute.
- **Customizing timeouts required for Reclaim Space Operation**
  Depending on the RBD volume size and its data pattern, Reclaim Space Operation might fail with the `context deadline exceeded` error. You can avoid this by increasing the timeout value.

Enabling reclaim space operation using Annotating Persistent Volume Claims

Enable reclaim space operations using Annotating Persistent Volume Claims.

About this task

Use this procedure to annotate **PersistentVolumeClaims** so that it can invoke the reclaim space operation automatically based on a given schedule.
Note:

- The schedule value is in the same format as the Kubernetes CronJobs which sets the and/or interval of the recurring operation request.
- Recommended schedule interval is **@weekly**. If the schedule interval value is empty or in an invalid format, then the default schedule value is set to **@weekly**.
- Minimum supported interval between each scheduled operation is at least 24 hours. For example, **@daily** (At 00:00 every day) or **0 3 * * *** (At 3:00 every day).
- Schedule the **ReclaimSpace** operation during off-peak, maintenance window, or the interval when the workload input/output is expected to be low.
- **ReclaimSpaceCronJob** is recreated when the **schedule** is modified. It is automatically deleted when the annotation is removed.

Procedure

1. Get the persistent volume claim (PVC) details, using the **pvc data-pvc** command.

   ```
   $ oc get pvc data-pvc
   ```

   For example:

   ```
   NAME       STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
   AGE
   data-pvc   Bound    pvc-f37b8582-4b04-4676-88dd-e1b95c6abf74   1Gi        RWO            ocs-storagecluster-ceph-rbd
   20h
   ```

2. Add annotation **reclaimspace.csiaddons.openshift.io/schedule=@monthly** to the PVC to create **reclaimspacecronjob**.
   For example:

   ```
   $ oc annotate pvc data-pvc "reclaimspace.csiaddons.openshift.io/schedule=@monthly"
   ```

   Output example:

   ```
   persistentvolumeclaim/data-pvc annotated
   ```

3. Verify that **reclaimspacecronjob** is created in the following format:
   <pvc-name>-xxxxxxx

   ```
   $ oc get reclaimspacecronjobs.csiaddons.openshift.io
   ```

   For example:

   ```
   NAME                      SCHEDULE     SUSPEND   ACTIVE   LASTSCHEDULE   AGE
   data-pvc-1642663516    @monthly                                  3s
   ```

4. Modify the schedule to run this job automatically.

   ```
   $ oc annotate pvc data-pvc "reclaimspace.csiaddons.openshift.io/schedule=@weekly" --overwrite=true
   ```

   For example:

   ```
   persistentvolumeclaim/data-pvc annotated
   ```

5. Verify that the schedule for **reclaimspacecronjob** has been modified.

   ```
   oc get reclaimspacecronjobs.csiaddons.openshift.io
   ```

   For example:

   ```
   NAME                   SCHEDULE     SUSPEND   ACTIVE   LASTSCHEDULE   AGE
   data-pvc-1642664617   @weekly                                  3s
   ```

Disabling reclaim space for a specific PersistentVolumeClaim

To disable reclaim space for a specific PersistentVolumeClaim (PVC), modify the associated **ReclaimSpaceCronJob** custom resource (CR).

Procedure

1. Identify the **ReclaimSpaceCronJob** CR for the PVC you want to disable reclaim space.
   For example:

   ```
   oc get reclaimspacecronjobs -o jsonpath='{range .items[?(@.spec.jobTemplate.spec.target.persistentVolumeClaim=="
   <PVC_NAME>")]}{.metadata.name}{"\n"}{end}'
   ```

   Replace **"<PVC_NAME>"** with the name of the PVC.

2. Disable the reclaim space from the **ReclaimSpaceCronJob** CR as follows:
   a. Update the **csiaddons.openshift.io/state** annotation from **"managed"** to **"unmanaged"**.
      For example:

```
oc annotate reclaimspacecronjobs <RECLAIMSPACECRONJOB_NAME> "csiaddons.openshift.io/state=unmanaged" --overwrite=true
```

Replace **<RECLAIMSPACECRONJOB_NAME>** with the **ReclaimSpaceCronJob** CR.

b. Add **suspend: true** under the **spec** field.
For example:

```
oc patch reclaimspacecronjobs <RECLAIMSPACECRONJOB_NAME> -p '{"spec": {"suspend": true}}' --type=merge
```

Enabling reclaim space operation using ReclaimSpaceJob

ReclaimSpaceJob is a namespaced custom resource designed to invoke reclaim space operation on the target volume.

About this task

This is a one time method that immediately starts the reclaim space operation. You have to repeat the creation of **ReclaimSpaceJob** CR to repeat the reclaim space operation when required.

Note:

- Recommended interval between the reclaim space operations is **weekly**.
- Ensure that the minimum interval between each operation is at least **24 hours**.
- Schedule the reclaim space operation during off-peak, maintenance window, or when the workload input/output is expected to be low.

Procedure

1. Create and apply the following custom resource for reclaim space operation:

```
apiVersion: csiaddons.openshift.io/v1alpha1
kind: ReclaimSpaceJob
metadata:
  name: sample-1
spec:
  target:
    persistentVolumeClaim: pvc-1
  timeout: 360
```

target
: Indicates the volume target on which the operation is going to perform.

persistentVolumeClaim
: Contains a string indicating the name of **PersistentVolumeClaim**.

backOfflimit
: Specifies the maximum number of retries before the reclaim space operation fails. If not specified, default value sets to 6. The maximum and minimum allowed value are 60 and 0.

retryDeadlineSeconds
: Specifies the duration in which the operation might retire in seconds and it is relative to the start time. The value must be a positive integer. The default value is

600 seconds and the allowed maximum value is **1800**seconds.

timeout
: Specifies the timeout in seconds for the **grpc** request sent to the CSI driver. If the timeout value is not specified, it defaults to the value of global reclaimspace timeout. Minimum allowed value for timeout is 60.

2. Delete the customer resource after completion of the operation.

Enabling reclaim space operation using ReclaimSpaceCronJob

ReclaimSpaceCronJob invokes the reclaim space operation based on the given schedule (daily, weekly, and so on). You have to create **ReclaimSpaceCronJob** one time only for a persistent volume claim. The **CSI-addons** controller creates a **ReclaimSpaceJob** at the requested time and interval with the schedule attribute.

About this task

Note:

- Recommended schedule interval is **@weekly**.
- Minimum interval between each scheduled operation should be at least 24 hours. For example, **@daily** (At 00:00 every day) or “0 3 * * *” (At 3:00 every day).
- Schedule the ReclaimSpace operation during off-peak, maintenance window, or the interval when workload input/output is expected to be low.

Procedure

1. Create and apply the following custom resource for reclaim space operation:

```
apiVersion: csiaddons.openshift.io/v1alpha1
kind: ReclaimSpaceCronJob
metadata:
  name: reclaimspacecronjob-sample
spec:
```

```
  jobTemplate:
    spec:
      target:
        persistentVolumeClaim: data-pvc
  schedule: '@weekly'
  concurrencyPolicy: Forbid
```

concurrencyPolicy
: Describes the changes when a new **ReclaimSpaceJob** is scheduled by the **ReclaimSpaceCronJob**, while a previous **ReclaimSpaceJob** is still running. The default **Forbid** prevents starting a new job whereas **Replace** can be used to delete the running job potentially in a failure state and create a new one.

failedJobsHistoryLimit
: Specifies the number of failed **ReclaimSpaceJobs** that are kept for troubleshooting.

jobTemplate
: Specifies the ReclaimSpaceJob.spec structure that describes the details of the requested **ReclaimSpaceJob** operation.

successfulJobsHistoryLimit
: Specifies the number of successful **ReclaimSpaceJob** operations.

schedule
: Sets the and/or interval of the recurring operation. For the schedule format, see API reference > Workloads APIs > CronJob in Red Hat OpenShift Container Platform product documentation.

2. Delete the **ReclaimSpaceCronJob** customer resource after completion of the operation or when the target PVC is deleted.

Customizing timeouts required for Reclaim Space Operation

Depending on the RBD volume size and its data pattern, Reclaim Space Operation might fail with the **context deadline exceeded** error. You can avoid this by increasing the timeout value.

About this task

The following example shows the failed status by inspecting **-o yaml** of the corresponding **ReclaimSpaceJob**:

Example

```
Status:
  Completion Time:  2023-03-08T18:56:18Z
  Conditions:
    Last Transition Time:  2023-03-08T18:56:18Z
    Message:               Failed to make controller request: context deadline exceeded
    Observed Generation:   1
    Reason:                failed
    Status:                True
    Type:                  Failed
  Message:                 Maximum retry limit reached
  Result:                  Failed
  Retries:                 6
  Start Time:              2023-03-08T18:33:55Z
```

You can also set custom timeouts at global level by creating the following **configmap**:

Example

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: csi-addons-config
  namespace: openshift-storage
data:
  "reclaim-space-timeout": "6m"
```

Restart the **csi-addons** operator pod.

```
oc delete po -n openshift-storage -l "app.kubernetes.io/name=csi-addons"
```

All Reclaim Space Operations started after the preceding configmap creation use the customized timeout.

Finding and cleaning stale subvolumes

This section describes how to find and clean stale subvolumes.

Before you begin

Important: Deleting stale subvolumes using the ODF CLI tool is a Technology Preview feature and is subject to Technology Preview limitations.
Download the OpenShift Data Foundation command-line interface (CLI) tool. With the Data Foundation CLI tool, you can effectively manage and troubleshoot your Data Foundation environment from a terminal. To get a compatible version and download the CLI tool, see customer portal.

About this task

In certain scenarios, subvolumes may become stale and lack an associated Kubernetes (k8s) reference. These orphaned subvolumes serve no functional purpose and can be safely removed to optimize storage utilization. You can identify and delete stale subvolumes using the ODF CLI tool. This process ensures that unused resources are

cleaned up, maintaining a healthy and efficient storage environment.

Procedure

1. Find the stale subvolumes by using the --stale flag with the **subvolume** command:

   ```
   odf subvolume ls --stale
   ```

   Example output:

   ```
   # Filesystem  Subvolume Subvolumegroup State
   # ocs-storagecluster-cephfilesystem csi-vol-427774b4-340b-11ed-8d66-0242ac110004 csi stale
   # ocs-storagecluster-cephfilesystem csi-vol-427774b4-340b-11ed-8d66-0242ac110005 csi stale
   ```

2. Delete the stale subvolumes:

   ```
   odf subvolume delete <subvolumes> <filesystem> <subvolumegroup>
   ```

   Replace *<subvolumes>* with a comma separated list of subvolumes from the output of the first command. The subvolumes must be of the same filesystem and subvolumegroup.

   Replace *<filesystem>* and *<subvolumegroup>* with the filesystem and subvolumegroup from the output of the first command.

   Example:

   ```
   odf subvolume delete csi-vol-427774b4-340b-11ed-8d66-0242ac110004,csi-vol-427774b4-340b-11ed-8d66-0242ac110005 ocs-storagecluster csi
   ```

   Example output:

   ```
   # Info: subvolume csi-vol-427774b4-340b-11ed-8d66-0242ac110004 deleted
   # Info: subvolume csi-vol-427774b4-340b-11ed-8d66-0242ac110004 deleted
   ```

Volume snapshots

Create, restore, and delete volume snapshots.

A volume snapshot is the state of the storage volume in a cluster at a particular point in time. These snapshots help to use storage more efficiently by not having to make a full copy each time and can be used as building blocks for developing an application.

Volume snapshot class allows an administrator to specify different attributes belonging to a volume snapshot object. The Fusion Data Foundation operator installs default volume snapshot classes depending on the platform in use. The operator owns and controls these default volume snapshot classes and they cannot be deleted or modified.

You can create many snapshots of the same persistent volume claim (PVC) but cannot schedule periodic creation of snapshots.

- For CephFS, you can create up to 100 snapshots per PVC.
- For RADOS Block Device (RBD), you can create up to 512 snapshots per PVC.

Note: Persistent Volume encryption now supports volume snapshots.

- **Creating volume snapshots**
  You can create a volume snapshot either from the Persistent Volume Claim (PVC) page or the Volume Snapshots page.
- **Restoring volume snapshots**
  When you restore a volume snapshot, a new Persistent Volume Claim (PVC) gets created. The restored PVC is independent of the volume snapshot and the parent PVC.
- **Deleting volume snapshots**
  You can delete a volume snapshot either from the Persistent Volume Claim (PVC) page or the Volume Snapshots page.

Creating volume snapshots

You can create a volume snapshot either from the Persistent Volume Claim (PVC) page or the Volume Snapshots page.

For a consistent snapshot, the PVC should be in *Bound* state and not be in use. Ensure to stop all I/O before taking the snapshot.
Note: Fusion Data Foundation only provides crash consistency for a volume snapshot of a PVC if a pod is using it. For application consistency, be sure to first tear down a running pod to ensure consistent snapshots or use any quiesce mechanism provided by the application to ensure it.

Creating volume snapshots from the Persistent Volume Claims page

1. From the OpenShift Web Console go to Storage > Persistent Volume Claims.
2. Create a volume snapshot in one of the following ways:
   - Beside the desired PVC, go to Action Menu > Create Snapshot.
   - Click the PVC for which you want to create the snapshot and click Actions > Create Snapshot.
3. Enter a Name for the volume snapshot.
4. Choose the Snapshot Class from the drop-down list.
5. Click Create.
   You are redirected to the Details page of the volume snapshot that is created.

Creating volume snapshots from the Volume Snapshots page

1. From the OpenShift Web Console go to Storage > Volume Snapshots.
2. In the Volume Snapshots page, click Create Volume Snapshot.
3. Choose the required Project from the drop-down list.
4. Choose the Persistent Volume Claim from the drop-down list.
5. Enter a Name for the snapshot.
6. Choose the Snapshot Class from the drop-down list.
7. Click Create.
   You are redirected to the Details page of the volume snapshot that is created.

Verifying volume snapshot creation

- Go to the Details page of the PVC and click the Volume Snapshots tab to see the list of volume snapshots. Verify that the new volume snapshot is listed.
- From the OpenShift Web Console go to Storage > Volume Snapshots. Verify that the new volume snapshot is listed.
- Wait for the volume snapshot to be in a *Ready* state.

Restoring volume snapshots

When you restore a volume snapshot, a new Persistent Volume Claim (PVC) gets created. The restored PVC is independent of the volume snapshot and the parent PVC.

A volume snapshot can be restored either the Persistent Volume Claim page or the Volume Snapshots page.

Restoring volume snapshots from the Persistent Volume Claims page

Important: You can restore volume snapshot from the Persistent Volume Claims page only if the parent PVC is present.

1. From the OpenShift Web Console go to Storage > Persistent Volume Claims.
2. Click the PVC name with the volume snapshot to restore a volume snapshot as a new PVC.
3. In the Volume Snapshots tab, next to the volume snapshot you want to restore, click Action Menu > Restore as new PVC.
4. Enter a name for the new PVC.
5. Select the Storage Class name.
6. Select the Access Mode of your choice.
7. For RBD, select Volume mode.
8. Click Restore.
   You are redirected to the new PVC details page.

Restoring volume snapshots from the Volume Snapshots page

1. From the OpenShift Web Console go to Storage > Volume Snapshots.
2. In the Volume Snapshots tab, next to the volume snapshot you want to restore, click Action Menu > Restore as new PVC.
3. Enter a name for the new PVC.
4. Select the Storage Class name.
5. Select the Access Mode of your choice.
6. For RBD, select Volume mode.
7. Click Restore.
   You are redirected to the new PVC details page.

Verifying volume snapshot restoration

- From the OpenShift Web Console go to Storage > Persistent Volume Claims. Verify that the new PVC is listed in the Persistent Volume Claims page.
- Wait for the PVC to be in a *Bound* state.

Deleting volume snapshots

You can delete a volume snapshot either from the Persistent Volume Claim (PVC) page or the Volume Snapshots page.

For deleting a volume snapshot, the volume snapshot class which is used in that particular volume snapshot should be present.

Deleting volume snapshots from the Persistent Volume Claims page

1. From the OpenShift Web Console go to Storage > Persistent Volume Claims.
2. Click the PVC name which has the volume snapshot that needs to be deleted.
3. In the Volume Snapshots tab, next to the volume snapshot you want to restore, click Action Menu > Delete Volume Snapshot.

Deleting volume snapshots from the Volume Snapshots page

1. From the OpenShift Web Console go to Storage > Volume Snapshots.
2. In the Volume Snapshots page, click Delete Volume Snapshot.

Verifying volume snapshot deletion

- Ensure that the deleted volume snapshot is not present in the Volume Snapshots tab of the PVC details page.
- From the OpenShift Web Console go to Storage > Volume Snapshots. Verify that the deleted volume snapshot is not listed.

Volume cloning

A clone is a duplicate of an existing storage volume that is used as any standard volume. You create a clone of a volume to make a point in time copy of the data. A persistent volume claim (PVC) cannot be cloned with a different size. You can create up to 512 clones per PVC for both CephFS and RADOS Block Device (RBD).

- **Creating a clone**
  Create a clone.

Creating a clone

Create a clone.

Before you begin

Source PVC must be in *Bound* state and must not be in use.
Note: Do not create a clone of a PVC if a Pod is using it. Doing so might cause data corruption because the PVC is not quiesced (paused).

Procedure

1. From the OpenShift Web Console go to Storage > Persistent Volume Claims.
2. Create a clone in one of the following ways:
   - Beside the desired PVC, go to Action Menu > Clone PVC.
   - Click the PVC for which you want to clone and click Actions > Clone PVC.
3. Enter a Name for the clone.
4. Select the access mode of your choice.
5. Enter the required size of the clone.
6. Select the storage class in which you want to create the clone.
   The storage class can be any RBD storage class and it need not necessarily be the same as the parent PVC.
7. Click Clone.
   You are redirected to the new PVC details page.
8. Wait for the cloned PVC status to become *Bound*.
   The cloned PVC is now available to be consumed by the pods. This cloned PVC is independent of its dataSource PVC.

Managing Container Storage Interface (CSI) component placements

You can set tolerations to run the Container Storage Interface (CSI) components on the **`infra`** nodes with custom taints.

About this task

Each cluster consists of multiple dedicated nodes such as **`infra`** and **`storage`** nodes. An **`infra`** node with a custom taint cannot use Fusion Data Foundation Persistent Volume Claims (PVCs) on the node unless the required tolerations are added. To use such nodes, configure tolerations to bring up **`csi-plugins`** on the nodes.

Procedure

1. Get the list of subscriptions.
   Command example:

   ```
   oc get subs -n openshift-storage
   ```

   Example output:

   ```
   NAME                                                                PACKAGE                  SOURCE             CHAN
   mcg-operator-stable-4.16-redhat-operators-openshift-marketplace     mcg-operator             redhat-operators   stab
   ocs-client-operator-stable-4.16-redhat-operators-openshift-marketplace   ocs-client-operator      redhat-operators   stab
   ocs-operator-stable-4.16-redhat-operators-openshift-marketplace     ocs-operator             redhat-operators   stab
   odf-csi-addons-operator-stable-4.16-redhat-operators-openshift-marketplace   odf-csi-addons-operator   redhat-operators   stab
   odf-operator                                                        odf-operator             redhat-operators   stab
   odf-prometheus-operator-stable-4.16-redhat-operators-openshift-marketplace   odf-prometheus-operator   redhat-operators   stab
   recipe-stable-4.16-redhat-operators-openshift-marketplace           recipe                   redhat-operators   stab
   rook-ceph-operator-stable-4.16-redhat-operators-openshift-marketplace   rook-ceph-operator       redhat-operators   stab
   ```

2. Edit the **`odf-operator`** subscriptions to add the required toleration to the **`odf-console`**, **`odf-operator-controller-manager`**, **`ocs-operator`**, **`rook-ceph-operator`**, **`csi-addons-controller-manager`**, **`ux-backend-server`**, and **`noobaa-operator`** pods.
   Command example:

   ```
   oc edit sub odf-operator -n openshift-storage
   ```

Toleration example:

```
spec:
  config:
    tolerations:
    - effect: NoSchedule
      key: xyz
      operator: Equal
      value: "true"
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      operator: Equal
      value: "true"
```

Remember: Save your changes before exiting the editor.

3. Edit the **StorageCluster** custom resource (CR) to add the desired toleration for the **Rook Ceph**, **Noobaa**, **csi-plugin**, **csi-provisioner**, **metrics-exporter**, **toolbox**, **ocs-provider-server**, and **blackbox-exporter** pods.
   To add tolerations to CSI pods, do not pass CSI-specific tolerations in the **StorageCluster** CR. Instead, follow these steps:

   a. List all CSI drivers that need tolerations for custom taints.
      Command example:

      ```
      oc get driver
      ```

      Example output:

      ```
      NAME                                     AGE
      openshift-storage.cephfs.csi.ceph.com    6d23h
      openshift-storage.nfs.csi.ceph.com       6d23h
      openshift-storage.rbd.csi.ceph.com       6d23h
      ```

   b. Edit the listed drivers to add tolerations for custom taints.
      Command example:

      ```
      oc edit driver <driver_name> -n openshift-storage
      ```

      Toleration example:

      ```
      spec
        controllerPlugin:
          tolerations:
          - effect: NoSchedule
            key: xyz
            operator: Equal
            value: "true"
          - effect: NoSchedule
            key: node.ocs.openshift.io/storage
            operator: Equal
            value: "true"
        nodePlugin:
          tolerations:
          - effect: NoSchedule
            key: xyz
            operator: Equal
            value: "true"
          - effect: NoSchedule
            key: node.ocs.openshift.io/storage
            operator: Equal
            value: "true"
          blackbox-exporter:
            tolerations:
            - effect: NoSchedule
              key: xyz
              operator: Equal
              value: "true"
            - effect: NoSchedule
              key: node.ocs.openshift.io/storage
              operator: Equal
              value: "true"
      ```

      Remember: Save your changes before exiting the editor.

Command example:

```
oc edit storagecluster ocs-storagecluster -n openshift-storage
```

Toleration example:

```
spec:
  placement:
    all:
      tolerations:
      - effect: NoSchedule
        key: xyz
        operator: Equal
        value: "true"
      - effect: NoSchedule
        key: node.ocs.openshift.io/storage
        operator: Equal
        value: "true"
    csi-plugin:
      tolerations:
      - effect: NoSchedule
        key: xyz
```

```
      operator: Equal
      value: "true"
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      operator: Equal
      value: "true"
  csi-provisioner:
    tolerations:
    - effect: NoSchedule
      key: xyz
      operator: Equal
      value: "true"
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      operator: Equal
      value: "true"
  mds:
    tolerations:
    - effect: NoSchedule
      key: xyz
      operator: Equal
      value: "true"
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      operator: Equal
      value: "true"
  metrics-exporter:
    tolerations:
    - effect: NoSchedule
      key: xyz
      operator: Equal
      value: "true"
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      operator: Equal
      value: "true"
  noobaa-core:
    tolerations:
    - effect: NoSchedule
      key: xyz
      operator: Equal
      value: "true"
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      operator: Equal
      value: "true"
  rgw:
    tolerations:
    - effect: NoSchedule
      key: xyz
      operator: Equal
      value: "true"
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      operator: Equal
      value: "true"
  toolbox:
    tolerations:
    - effect: NoSchedule
      key: xyz
      operator: Equal
      value: "true"
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      operator: Equal
      value: "true"
  api-server:
    tolerations:
    - effect: NoSchedule
      key: xyz
      operator: Equal
      value: "true"
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      operator: Equal
      value: "true"
```

Remember: Save your changes before exiting the editor.

2-way replication with CephFS

To reduce storage load with the Ceph File System (CephFS) when data resiliency is not a primary concern, you can choose to use 2-way replication (replica-2). It reduces the amount of storage space that is used and decreases the level of fault tolerance.

There are two ways to use replica-2 for CephFS:

1. Edit the existing default pool to replica-2 and use it with the default CephFS storage class.
2. Add an additional CephFS data pool with replica-2.

- **Editing the existing default CephFS data pool to replica-2**
  Use this procedure to edit the existing default CephFS pool to replica-2 and use it with the default CephFS storage class.

- **Adding an additional CephFS data pool with replica-2**
  Use this procedure to add an additional CephFS data pool with replica-2.

Editing the existing default CephFS data pool to replica-2

Use this procedure to edit the existing default CephFS pool to replica-2 and use it with the default CephFS storage class.

Procedure

1. Patch the storagecluster to change default CephFS data pool to replica-2.

```
oc patch storagecluster ocs-storagecluster -n openshift-storage --type json --patch '[{ "op": "replace", "path": "/spec/managedResources/cephFilesystems/dataPoolSpec/replicated/size", "value": 2 }]'
```

2. Run the following command to verify the CephFS details:

```
oc get cephfilesystem ocs-storagecluster-cephfilesystem -o=jsonpath='{.spec.dataPools}' | jq
```

   Example output:

```
[
  {
    "application": "",
    "deviceClass": "ssd",
    "erasureCoded": {
      "codingChunks": 0,
      "dataChunks": 0
    },
    "failureDomain": "zone",
    "mirroring": {},
    "quotas": {},
    "replicated": {
      "replicasPerFailureDomain": 1,
      "size": 2,
      "targetSizeRatio": 0.49
    },
    "statusCheck": {
      "mirror": {}
    }
  }
]
```

3. Run the following command to check the pool details:

```
ceph osd pool ls | grep filesystem
```

   Example output:

```
ocs-storagecluster-cephfilesystem-metadata
ocs-storagecluster-cephfilesystem-data0
```

Adding an additional CephFS data pool with replica-2

Use this procedure to add an additional CephFS data pool with replica-2.

Before you begin

Ensure that you are logged into the OpenShift® Container Platform web console and Fusion Data Foundation cluster is in ready state.

Procedure

1. Go to Storage > StorageClasses.
2. Click Create StorageClass.
3. Select the CephFS Provisioner.
4. In the Storage Pool tab, click Create new storage pool.
5. On the Create Storage Pool dialog, do as follows:
   a. Select the required volume type.
   b. Add a pool name.
   c. In the Data protection policy field, select 2-way Replication.
   d. Click Create.
6. In the Storage Class creation form, choose the newly created storage pool.
7. Confirm the Storage Class creation.
8. Do the following steps to verify the details of the newly created storage system:
   a. Go to Storage > Data Foundation.
   b. In the Data Foundation window, go to the Storage Systems tab and select the new storage system.
   c. Check whether the Details tab of the storage system reflects the correct volume and device types.

Creating exports using NFS

Create exports using NFS that can then be accessed externally from the Fusion Data Foundation cluster.

- **Enabling the NFS feature**
  To use the NFS feature, it needs to be enabled in the cluster.
- **Creating NFS exports**
  NFS exports are created by creating a Persistent Volume Claim (PVC) against the `ocs-storagecluster-ceph-nfs` StorageClass.
- **Consuming NFS exports in-cluster**
  Kubernetes application pods can consume NFS exports created by mounting a previously created PVC.
- **Consume NFS exports externally from the cluster**
  NFS clients outside of the cluster can mount NFS exports created by a previously-created PVC.

Enabling the NFS feature

To use the NFS feature, it needs to be enabled in the cluster.

Before you begin

Ensure you have the following:

- Fusion Data Foundation is installed and running in the **openshift-storage** namespace.
- The Fusion Data Foundation installation includes a CephFilesystem.

Procedure

- Run the following command to enable the NFS feature:

  ```
  oc --namespace openshift-storage patch storageclusters.ocs.openshift.io ocs-storagecluster --type merge --patch '{"spec": {"nfs":{"enable": true}}}'
  ```

- Specify a custom NFS Ganesha server for NFS StorageClasses.
  You can optionally set `externalEndpoint` in `StorageCluster.spec.nfs` to define the externally resolvable IP address or DNS name of the NFS Ganesha server to be used by NFS StorageClass.

  Note: When configured, internal clients also use this address instead of the default NFS service name.

  ```
  oc --namespace openshift-storage patch storageclusters.ocs.openshift.io ocs-storagecluster \
    --type=merge \
    --patch '{"spec":{"nfs":{"externalEndpoint":"<VALUE>"}}}'
  ```

What to do next

NFS installation and configuration is complete when the following conditions are met:

- The CephNFS resource named `ocs-storagecluster-cephnfs` has a status of *Ready*.
- Check all `csi-nfsplugin-*` pods are running:

  ```
  oc -n openshift-storage describe cephnfs ocs-storagecluster-cephnfs
  ```

  ```
  oc -n openshift-storage get pod | grep csi-nfsplugin
  ```

  Output has multiple pods. For example:

  ```
  csi-nfsplugin-47qwq                                               2/2     Running  0  10s
  csi-nfsplugin-77947                                               2/2     Running  0  10s
  csi-nfsplugin-ct2pm                                               2/2     Running  0  10s
  csi-nfsplugin-provisioner-f85b75fbb-2rm2w                         2/2     Running  0  10s
  csi-nfsplugin-provisioner-f85b75fbb-8nj5h                         2/2     Running  0  10s
  ```

Creating NFS exports

NFS exports are created by creating a Persistent Volume Claim (PVC) against the `ocs-storagecluster-ceph-nfs` StorageClass.

NFS PVCs can be created either by using a YAML file or from the OpenShift Container Platform web console.

Create NFS PVC using a YAML file

Use the following example PVC, where *<desired_name>* specifies a name for the PVC, for example, `my-nfs-export`.
Note: `volumeMode: Block` does not work for NFS volumes.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: <desired_name>
spec:
```

```
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 1Gi
storageClassName: ocs-storagecluster-ceph-nfs
```

The export is created once the PVC reaches the *Bound* state.

Create NFS PVCs from the OpenShift Container Platform web console

Ensure that you are logged into the OpenShift Container Platform web console and the NFS feature is enabled for the storage cluster.

1. From the OpenShift Web Console go to Storage > Persistent Volume Claims.
2. Set the Project to **openshift-storage**.
3. Click Create PersistentVolumeClaim and fill in the form.
    a. Specify the Storage Class: **ocs-storagecluster-ceph-nfs**.
    b. Specify the PVC Name. For example, **my-nfs-export**.
    c. Select the required Access Mode.
    d. Specify a Size as per application requirement.
    e. Select Volume mode: **Filesystem**.
    Note: **Block** mode is not supported for NFS PVCs.
    f. Click Create and wait until the PVC is in a *Bound* status.

Consuming NFS exports in-cluster

Kubernetes application pods can consume NFS exports created by mounting a previously created PVC.

Mount the PVCs either by using a YAML file or from the OpenShift Container Platform web console.

Mounting a PVC using a YAML file

This example uses the same example pod, used in the PVC creation example in Create NFS PVC using a YAML file, where *<pvc_name>* specifies the PVC you have previously created, for example, **my-nfs-export**.

```
apiVersion: v1
kind: Pod
metadata:
 name: nfs-export-example
spec:
 containers:
   - name: web-server
     image: nginx
     volumeMounts:
       - name: nfs-export-pvc
         mountPath: /var/lib/www/html
 volumes:
   - name: nfs-export-pvc
     persistentVolumeClaim:
       claimName: <pvc_name>
       readOnly: false
```

Mounting a PVC from the OpenShift Container Platform web console

1. From the OpenShift Web Console go to Workloads > Pods.
2. Click Create Pod to create a new application pod.
3. From the metadata section add a name. For example, `nfs-export-example`, with namespace as `openshift-storage`.
4. Under the **spec:** section, add **containers:** section with **image** and **volumeMounts** sections.

```
apiVersion: v1
kind: Pod
metadata:
 name: nfs-export-example
 namespace: openshift-storage
spec:
 containers:
   - name: web-server
     image: nginx
     volumeMounts:
       - name: <volume_name>
         mountPath: /var/lib/www/html
```

For example:

```
apiVersion: v1
kind: Pod
metadata:
 name: nfs-export-example
 namespace: openshift-storage
spec:
 containers:
   - name: web-server
     image: nginx
     volumeMounts:
```

```
      - name: nfs-export-pvc
        mountPath: /var/lib/www/html
```

5. Under the `spec:` section, add `volumes:` section to add the NFS PVC as a volume for the application pod.

```
volumes:
  - name: <volume_name>
    persistentVolumeClaim:
      claimName: <pvc_name>
```

For example:

```
volumes:
  - name: nfs-export-pvc
    persistentVolumeClaim:
      claimName: my-nfs-export
```

Consume NFS exports externally from the cluster

NFS clients outside of the cluster can mount NFS exports created by a previously-created PVC.

Procedure

1. After the `nfs` flag is enabled, singe-server CephNFS is deployed by Rook. You need to fetch the value of the `ceph_nfs` field for the `nfs-ganesha` server. The `ceph_nfs` field for the `nfs-ganesha` server are used in step 2.

```
oc get pods -n openshift-storage | grep rook-ceph-nfs
```

```
oc describe pod  <name of the rook-ceph-nfs pod> | grep ceph_nfs
```

For example:

```
oc describe pod rook-ceph-nfs-ocs-storagecluster-cephnfs-a-7bb484b4bf-bbdhs | grep ceph_nfs
  ceph_nfs=my-nfs
```

2. Expose the NFS server outside of the Fusion Data Foundation cluster by creating a Kubernetes LoadBalancer Service. This example creates a LoadBalancer Service and references the NFS server created by Fusion Data Foundation. Replace *<my-nfs>* with the value you got in step 1.

```
apiVersion: v1
kind: Service
metadata:
 name: rook-ceph-nfs-ocs-storagecluster-cephnfs-load-balancer
 namespace: openshift-storage
spec:
 ports:
   - name: nfs
     port: 2049
 type: LoadBalancer
 externalTrafficPolicy: Local
 selector:
   app: rook-ceph-nfs
   ceph_nfs: <my-nfs>
   instance: a
```

3. Collect connection information.
The information external clients need to connect to an export comes from the Persistent Volume (PV) created for the PVC, and the status of the LoadBalancer Service created in step 2.

   a. Get the share path from the PV.
      i. Get the name of the PV associated with the NFS export's PVC:

```
oc get pvc <pvc_name> --output jsonpath='{.spec.volumeName}'
pvc-39c5c467-d9d3-4898-84f7-936ea52fd99d
```

Replace *<pvc_name>* with your own PVC name. For example:

```
oc get pvc pvc-39c5c467-d9d3-4898-84f7-936ea52fd99d --output jsonpath='{.spec.volumeName}'
pvc-39c5c467-d9d3-4898-84f7-936ea52fd99d
```

      ii. Use the PV name obtained previously to get the NFS export's share path:

```
oc get pv pvc-39c5c467-d9d3-4898-84f7-936ea52fd99d --output jsonpath='{.spec.csi.volumeAttributes.share}'
/0001-0011-openshift-storage-0000000000000001-ba9426ab-d61b-11ec-9ffd-0a580a800215
```

   b. Get an ingress address for the NFS server.
   A service's ingress status may have multiple addresses. Choose the one desired to use for external clients. In the following example, there is only a single address: the host name `ingress-id.somedomain.com`.

```
oc -n openshift-storage get service rook-ceph-nfs-ocs-storagecluster-cephnfs-load-balancer --output
jsonpath='{.status.loadBalancer.ingress}'
[{"hostname":"ingress-id.somedomain.com"}]
```

4. Connect the external client using the share path and ingress address from the previous steps.
The following example mounts the export to the client's directory path /export/mount/path:

```
mount -t nfs4 -o proto=tcp ingress-id.somedomain.com:/0001-0011-openshift-storage-0000000000000001-ba9426ab-d61b-11ec-
9ffd-0a580a800215 /export/mount/path
```

If this does not work immediately, it could be that the Kubernetes environment is still taking time to configure the network resources to allow ingress to the NFS server.

Annotating encrypted RBD storage classes

This section provides you with information on annotating encrypted RBD storage classes.

When the OpenShift console creates a RADOS block device (RBD) storage class with encryption enabled, the annotation is set automatically. However, you need to add the annotation, `cdi.kubevirt.io/clone-strategy=copy` for any of the encrypted RBD storage classes that were previously created before updating to the Fusion Data Foundation. This enables customer data integration (CDI) to use host-assisted cloning instead of the default smart cloning.

The keys used to access an encrypted volume are tied to the namespace where the volume was created. When cloning an encrypted volume to a new namespace, such as, provisioning a new OpenShift Virtualization virtual machine, a new volume must be created and the content of the source volume must then be copied into the new volume. This behavior is triggered automatically if the storage class is properly annotated.

Enabling faster client IO or recovery IO during OSD backfill

Use this procedure for faster recovery during OSD backfill.

Before you begin

- Download the `odf-cli` from the customer portal

About this task

During a maintenance window, you may want to favor either client IO or recovery IO. Favoring recovery IO over client IO significantly reduces OSD recovery time. The valid recovery profile options are `balanced`, `high_client_ops`, and `high_recovery_ops`. Set the recovery profile using the following procedure.

Procedure

1. Check the current recovery profile:

   ```
   odf get recovery-profile
   ```

2. Modify the recovery profile:

   ```
   odf set recovery-profile <option>
   ```

   Replace `option` with either `balanced`, `high_client_ops`, or `high_recovery_ops`.
3. Verify the updated recovery profile:

   ```
   odf get recovery-profile
   ```

Setting Ceph OSD full thresholds

You can set Ceph Object Storage Daemon (OSD) full thresholds by using the OpenShift Data Foundation command-line interface (CLI) tool or by updating the StorageCluster Custom Resource (CR).

- **Setting Ceph OSD full thresholds by using the ODF CLI tool**
  Use this procedure to set Ceph OSD full thresholds temporarily by using the ODF CLI tool.
- **Setting Ceph OSD full thresholds by updating the StorageCluster CR**
  Use this procedure to override the default settings and to set Ceph OSD full thresholds by updating the StorageCluster CR.

Setting Ceph OSD full thresholds by using the ODF CLI tool

Use this procedure to set Ceph OSD full thresholds temporarily by using the ODF CLI tool.

Prerequisites

Whenever the cluster reaches its full state, it is required to immediately increase the thresholds. With the ODF CLI tool, you can effectively manage and troubleshoot your Fusion Data Foundation environment from a terminal. For more information about the compatible version details and to download the CLI, see customer portal.

Procedure

Use the **set** command to adjust Ceph full thresholds. The **set** command supports the **full**, **backfillfull**, and **nearfull** subcommands. See the following examples for how to use each subcommand.

full

Use this subcommand to update the full ratio of Ceph OSDs when Ceph prevents the IO operation on OSDs that reached the specified capacity. The default value is *0.85*.

Important: Do not set the full ratio value of OSDs too close to *1.0* because the cluster becomes unrecoverable if the OSDs are full and have nowhere to grow.
For example, run the following command to set the full ratio to *0.90*:

```
$ odf set full 0.90
```

backfillfull

Use this subcommand to update the backfillfull ratio of Ceph OSDs when Ceph denies backfilling to the OSDs that reached the specified capacity. The default value is *0.80*.

Important: If the value is set too close to *1.0*, the OSDs become full and prevent the cluster from backfilling.
For example, run the following command to set the backfillfull ratio to *0.85*:

```
$ odf set backfillfull 0.85
```

nearfull

Use this subcommand to update the nearfull ratio of Ceph OSDs when the cluster reaches the specified capacity and Ceph returns the nearfull message of OSDs. The default value is *0.75*.

For example, run the following command to set the nearfull ratio to *0.80*:

```
$ odf set nearfull 0.80
```

Setting Ceph OSD full thresholds by updating the StorageCluster CR

Use this procedure to override the default settings and to set Ceph OSD full thresholds by updating the StorageCluster CR.

Procedure

You can update the StorageCluster CR to change the settings for *fullRatio*, *backfillFullRatio*, and *nearFullRatio*.

fullRatio

Use this argument to update full ratio of Ceph OSDs when Ceph prevents the IO operation on OSDs that reached the specified capacity. The default value is *0.85*.

Important: Do not set the full ratio value of OSDs too close to *1.0* because the cluster becomes unrecoverable if the OSDs are full and have nowhere to grow.
For example, run the following command to set the *fullRatio* to *0.90*:

```
 oc patch storagecluster ocs-storagecluster -n openshift-storage --type json --patch '[{ "op": "replace", "path":
"/spec/managedResources/cephCluster/fullRatio", "value": 0.90 }]'
```

backfillFullRatio

Use this argument to update the backfillFull ratio of Ceph OSDs when Ceph denies backfilling to the OSDs that reached the specified capacity. The default value is *0.80*.

Important: If the value is set too close to *1.0*, the OSDs become full and prevent the cluster from backfilling.
For example, run the following command to set the *backfillFullRatio* to *0.85*:

```
 oc patch storagecluster ocs-storagecluster -n openshift-storage --type json --patch '[{ "op": "replace", "path":
"/spec/managedResources/cephCluster/backfillFullRatio", "value": 0.85 }]'
```

nearFullRatio

Use this subcommand to update the nearFull ratio of Ceph OSDs when the cluster reaches the specified capacity and Ceph returns the nearfull message of OSDs. The default value is *0.75*.

For example, run the following command to set the *nearFullRatio* to *0.80*:

```
 oc patch storagecluster ocs-storagecluster -n openshift-storage --type json --patch '[{ "op": "replace", "path":
"/spec/managedResources/cephCluster/nearFullRatio", "value": 0.80 }]'
```

Configuring Ceph target size ratios

Learn how to configure **targetSizeRatio** values to optimize Ceph storage distribution across block, filesystem, and object pools according to workload requirements.

Depending on the cluster usage and how you expect to fill the cluster among the three types of storage: block, shared filesystem, and object storage, you can set a value to the **targetSizeRatio** parameter. The target size ratio is a relative value that is used for balancing the data across pools. The sum of all pool target size ratios need not add up to 1 but it can be greater or lesser than 1 and Ceph adjusts the values accordingly. However, each value should be a nonzero value even though it is very small.

Note: Target size ratio does not mean that there is an immediate change in the placement group (PG) number of pools.

Procedure

- To optimize for block storage, run the following commands and set the target ratios:

  In the following example, block storage is optimized for 80%.

```
oc patch storagecluster ocs-storagecluster -n openshift-storage --type merge -p '
{
  "spec": {
    "managedResources": {
      "cephBlockPools": {
        "poolSpec": {
          "replicated": {
            "size": 3,
            "targetSizeRatio": 0.8
          }
        }
      }
    }
  }
}'
```

```
oc patch storagecluster ocs-storagecluster -n openshift-storage --type merge -p '
{
  "spec": {
    "managedResources": {
      "cephFilesystems": {
        "dataPoolSpec": {
          "replicated": {
            "size": 3,
            "targetSizeRatio": 0.2
          }
        }
      }
    }
  }
}'
```

```
ceph osd pool autoscale-status
POOL                                          SIZE  TARGET SIZE  RATE  RAW CAPACITY   RATIO  TARGET RATIO  EFFECTIVE RATIO
BIAS  PG_NUM  NEW PG_NUM  AUTOSCALE  BULK
.mgr                                        580.0k                3.0         6144G  0.0000
1.0      32              on         False
ocs-storagecluster-cephblockpool            216.7M                3.0         6144G  0.8000        0.8000           0.8000
1.0     256              on         False
ocs-storagecluster-cephfilesystem-metadata  55030                 3.0         6144G  0.0000
4.0      32              on         False
ocs-storagecluster-cephfilesystem-data0         0                 3.0         6144G  0.2000        0.2000           0.2000
1.0     128          32  on         False
```

- To optimize storage in the filesystem, run the following commands:

  In the following example, the filesystem is optimized to 98%. Block pool `targetSizeRatio`: 0.01/0.02 Filesystem `targetSizeRatio`: 0.98 Object store `targetSizeRatio`: 0.01 (if present)

```
oc patch storagecluster ocs-storagecluster -n openshift-storage --type merge -p '
{
  "spec": {
    "managedResources": {
      "cephBlockPools": {
        "poolSpec": {
          "replicated": {
            "size": 3,
            "targetSizeRatio": 0.02
          }
        }
      }
    }
  }
}'
```

```
oc patch storagecluster ocs-storagecluster -n openshift-storage --type merge -p '
{
  "spec": {
    "managedResources": {
      "cephFilesystems": {
        "dataPoolSpec": {
          "replicated": {
            "size": 3,
            "targetSizeRatio": 0.98
          }
        }
      }
    }
  }
}'
```

```
ceph osd pool autoscale-status
POOL                                          SIZE  TARGET SIZE  RATE  RAW CAPACITY   RATIO  TARGET RATIO  EFFECTIVE RATIO
BIAS  PG_NUM  NEW PG_NUM  AUTOSCALE  BULK
.mgr                                        580.0k                3.0         6144G  0.0000
1.0      32              on         False
ocs-storagecluster-cephblockpool            216.9M                3.0         6144G  0.0909        0.0200           0.0909
1.0     256          32  on         False
```

```
ocs-storagecluster-cephfilesystem-metadata  59126                 3.0         6144G  0.0000
4.0      32                 on         False
ocs-storagecluster-cephfilesystem-data0         0                 3.0         6144G  0.9091        0.2000           0.9091
1.0      32         256  on         False
```

- To optimize storage in `cephobjectstore`, run the following command:

  In the following example, the `cephobjectstore` is optimized to 80%.

```
oc patch storagecluster ocs-storagecluster -n openshift-storage --type merge -p '
{
  "spec": {
    "managedResources": {
      "cephObjectStores": {
        "dataPoolSpec": {
          "replicated": {
            "size": 3,
            "targetSizeRatio": 0.80
          }
        }
      }
    }
  }
}'
```

- To edit target size ratios of a custom `cephblockpool` data pool, run the following command:

```
oc patch cephblockpool ceph1-pool-block --type=json -p '[{"op": "add", "path": "/spec/replicated/targetSizeRatio",
"value":0.99}]
```

- To edit target size ratios of custom filesystem data pools, edit the `ocs-storagecluster` YAML under `managedResources` > `cephFilesystems` > `additionalDataPools` as follows:

```
      additionalDataPools:
      - compressionMode: aggressive
        name: pool-1
        replicated:
          size: 3
          targetSizeRatio: 0.80:
```

Configuring Ceph monitor failover timeout

Configure the Ceph monitor failover timeout to control when the system replaces an unhealthy monitor, providing flexibility during maintenance operations and improving cluster stability.

The monitor failover timeout setting gives cluster administrators precise control over Ceph monitor (mon) failover behavior.

By default, the `rook-ceph` operator replaces a down monitor after a fixed 10 minute interval to maintain quorum. While this supports availability, it can sometimes be too aggressive triggering failovers during short, expected outages, or too conservative, increasing the risk of quorum loss if another monitor fails during that window.

The configurable timeout parameters allow administrators to adjust this behavior or disable automatic monitor failover entirely. This flexibility is particularly useful during maintenance operations, such as node drains or reboots, when temporarily preventing monitor failover helps avoid unnecessary disruption. This provides more flexibility and control over monitor recovery behavior, simplifying maintenance, and improving stability during planned operations.

Parameters for monitor health check

Table 1. Parameters for monitor health check

Parameter	Description	Default	Comments
disabled	Boolean flag to enable or disable the monitor health check/failover mechanism.	false (enabled)	By default, automatic monitor failover is enabled. **Set this parameter to `true` only if you want to disable the monitor failover.**
interval	Specifies how long to wait between iterations	45s	-
timeout	Specifies how long the operator should wait for an unhealthy Ceph monitor to recover before triggering a failover.	10m	Default is 15m for IBM Cloud

These parameters are exposed in the `StorageCluster` CR, eliminating the need to configure them in the `CephCluster` CR manually. This simplifies management and keeps configurations consistent across cluster updates.

Example configuration

The following is an example of configuring the monitor failover timeout and disabling it directly in the `StorageCluster` CR:

```
apiVersion: ocs.openshift.io/v1
kind: StorageCluster
metadata:
  name: ocs-storagecluster
  namespace: openshift-storage
spec:
  managedResources:
    cephCluster:
      healthCheck:
        daemonHealth:
          mon:
            disabled: false # Ensure automatic monitor failover is enabled (default)
```

```
            interval: 30s   # Wait 30 seconds between iterations
            timeout: 15m    # Wait 15 minutes before failing over a down monitor
```

In this example, the failover timeout is increased to 15 minutes (from the default 10), and failover remains enabled.

To disable failover entirely, set **disabled: true** in the **StorageCluster** CR.

Backup and recovery of Multicloud Object Gateway metadata database (NooBaa DB)

Configure automated and on-demand backups for the Multicloud Object Gateway metadata database to maintain data resilience and improve recovery options.

The Multicloud Object Gateway supports automated and on-demand database backups to help maintain data resilience and improve recovery options. You can configure backup frequency during deployment by selecting the frequency, such as **none**, **daily**, **weekly**, or **monthly**. Also, you can define the number of retained backups (1–12). The Object dashboard displays the time of the most recent backup. In addition to scheduled backups, you can initiate on-demand backups through the MCG CLI or a dedicated custom resource. Backup schedules and retention settings can also be modified post-deployment through YAML-based configuration updates.

- **Enabling backup of Multicloud Object Gateway metadata database using storagecluster CR**
  Enable database backup for the Multicloud Object Gateway by configuring the **dbBackup** parameter in the storagecluster custom resource.
- **Enabling backup of Multicloud Object Gateway metadata database using MCG CLI**
  Initiate an on-demand backup of the Multicloud Object Gateway metadata database using the MCG command-line interface.
- **Recovering Multicloud Object Gateway metadata database from backup**
  Restore the Multicloud Object Gateway metadata database from a volume snapshot backup by configuring the storagecluster custom resource and triggering the recovery process.

Enabling backup of Multicloud Object Gateway metadata database using storagecluster CR

Enable database backup for the Multicloud Object Gateway by configuring the **dbBackup** parameter in the storagecluster custom resource.

Procedure

To enable DB backup using the storagecluster CR, configure **dbBackup** in the CR under **multiCloudGateway**:

```
multiCloudGateway:
  dbBackup:
    schedule: [cron format]
    volumeSnapshot:
      maxSnapshots: [integer]# number of snapshots to retain
      volumeSnapshotClass: [the vol snapshot class] # for ceph rbd ocs-storagecluster-rbdplugin-snapclass
```

- The operator creates a **scheduledbackups.postgresql.cnpg.noobaa.io** object.
- According to the schedule, the CNPG operator creates a **backups.postgresql.cnpg.noobaa.io** CR that triggers a volume snapshot of the secondary instance.

Enabling backup of Multicloud Object Gateway metadata database using MCG CLI

Initiate an on-demand backup of the Multicloud Object Gateway metadata database using the MCG command-line interface.

Procedure

To manually backup using MCG CLI, use the command:

```
noobaa system db-backup [--name custom-name]
```

MCG CLI creates a **backups.postgresql.cnpg.noobaa.io** with the given name, or by default **noobaa-db-pg-cluster-backup-[timestamp]**. The CLI command waits for the volume snapshot to be ready before exiting. Timeout is 5 minutes.

Recovering Multicloud Object Gateway metadata database from backup

Restore the Multicloud Object Gateway metadata database from a volume snapshot backup by configuring the storagecluster custom resource and triggering the recovery process.

Procedure

1. To recover the database from backup, set the **dbRecovery** property in the storagecluster CR:

   ```
   multiCloudGateway:
     dbRecovery:
       volumeSnapshotName: [volume snapshot to recover]
   ```

   Note: The configured volume snapshot is excluded from the retention policy and is not deleted as long as it is set in the CR.
2. To trigger the recovery, manually delete the cluster CR:

```
oc delete clusters.postgresql.cnpg.noobaa.io noobaa-db-pg-cluster
```

After the deletion:

a. NooBaa operator initiates the creation of a new cluster, recovering from the volume snapshot.
b. `noobaa-core` and `noobaa-operator` pods are stopped.
c. DB cluster is created.
d. Core and endpoints are restarted and the system becomes ready.

Managing hybrid and multicloud resource

Use this information to learn how to manage hybrid and multicloud resources.

The Multicloud Object Gateway (MCG) is a lightweight object storage service for OpenShift, allowing users to start small and then scale as needed on-premise, in multiple clusters, and with cloud-native storage.

- **Accessing the Multicloud Object Gateway with your applications**
  You can access the object service with any application targeting AWS S3 or code that uses AWS S3 Software Development Kit (SDK). Applications need to specify the Multicloud Object Gateway (MCG) endpoint, an access key, and a secret access key. You can use your terminal or the MCG command-line interface (CLI) to retrieve this information.
- **Adding storage resources for hybrid or Multicloud**
  Understand how to add and edit storage resources for hybrid or Multicloud.
- **Creating and managing buckets using the object browser**
  The object browser in the OpenShift Web Console enables you to create, view, and manage buckets and objects across different storage endpoints, including Multicloud Object Gateway (MCG) and internal mode RADOS Gateway (RGW) object storage.
- **Managing namespace buckets**
  Namespace buckets let you connect data repositories on different providers together, so you can interact with all of your data through a single unified view. Add the object bucket associated with each provider to the namespace bucket, and access your data through the namespace bucket to see all of your object buckets at once. This lets you write to your preferred storage provider while reading from multiple other storage providers, greatly reducing the cost of migrating to a new storage provider. Use this information to add namespace buckets using command-line interface, YAML, and user interface and for information about sharing and accessing legacy data.
- **Securing Multicloud Object Gateway**
  Ensure that the Multicloud Object Gateway is secure.
- **Mirroring data for hybrid and Multicloud buckets**
  Create bucket classes to mirror data for hybrid and Multicloud buckets. Mirroring data can be setup by using the OpenShift UI, YAML, or MCG command-line interface.
- **Bucket policies in the Multicloud Object Gateway**
  Fusion Data Foundation supports AWS S3 bucket policies. Bucket policies allow you to grant users access permissions for buckets and the objects in them. Use the information in this section to understand how to use bucket policies in Multicloud Object Gateway.
- **Bucket notification in Multicloud Object Gateway**
  With bucket notification, you can send a notification to an external server whenever there is an event in the bucket.
- **Multicloud Object Gateway bucket replication**
  Data replication from one Multicloud Object Gateway (MCG) bucket to another MCG bucket provides higher resiliency and better collaboration options. These buckets can be either data buckets or namespace buckets backed by any supported storage solution (S3, Azure, and so on). Replicate a bucket to another bucket and set replication policies using command-line interface and YAML.
- **Object Bucket Claim**
  An Object Bucket Claim can be used to request an S3 compatible bucket backend for your workloads. An object bucket claim creates a new bucket and an application account in NooBaa with permissions to the bucket, including a new access key and secret access key. The application account is allowed to access only a single bucket and can't create new buckets by default. Use this information to create, view, and delete object bucket claims.
- **Caching policy for object buckets**
  A cache bucket is a namespace bucket with a hub target and a cache target. The hub target is an S3 compatible large object storage bucket. The cache bucket is the local Multicloud Object Gateway bucket. You can create a cache bucket that caches an AWS bucket or an IBM COS bucket.
- **Lifecycle bucket configuration in Multicloud Object Gateway**
  Multicloud Object Gateway (MCG) lifecycle provides a way to reduce storage costs due to accumulated data objects.
- **Scaling Multicloud Object Gateway performance**
  The Multicloud Object Gateway (MCG) performance may vary from one environment to another. In some cases, specific applications require faster performance which can be addressed by scaling S3 endpoints.
- **Accessing the RADOS Object Gateway S3 endpoint**
  Users can access the RADOS Object Gateway (RGW) endpoint directly. The RGW route is created by default and is named `rook-ceph-rgw-ocs-storagecluster-cephobjectstore`.
- **Using TLS certificates for applications accessing RGW**
  Fetch TLS certificates stored as Kubernetes secrets and configure applications to use them when accessing the RADOS Object Gateway (RGW) over a secure connection.
- **Using the security token service of Multicloud Object Gateway to assume the role of another user**
  Multicloud Object Gateway (MCG) supports for a security token service (STS) similar to the one provided by Amazon Web Services. This STS of the MCG authorizes a user to assume the role of another user.

Accessing the Multicloud Object Gateway with your applications

You can access the object service with any application targeting AWS S3 or code that uses AWS S3 Software Development Kit (SDK). Applications need to specify the Multicloud Object Gateway (MCG) endpoint, an access key, and a secret access key. You can use your terminal or the MCG command-line interface (CLI) to retrieve this information.

Accessing the MCG buckets using the virtual-hosted style

If the client application tries to access `https://<bucket-name>.s3-openshift-storage.apps.mycluster-cluster.qe.rh-ocs.com`, a DNS entry is needed for **`mcg-test-bucket.s3-openshift-storage.apps.mycluster-cluster.qe.rh-ocs.com`** to point to the S3 Service.

<bucket-name> is the name of the MCG bucket, for example: **`https://mcg-test-bucket.s3-openshift-storage.apps.mycluster-cluster.qe.rh-ocs.com`**.

Important: Ensure that you have a DNS entry in order to point the client application to the MCG buckets using the virtual-hosted style.

- **Accessing the Multicloud Object Gateway from the terminal**
  Use this information to access the Multicloud Object Gateway from the terminal.
- **Accessing the Multicloud Object Gateway from the MCG command-line interface**
  Use this information to access the Multicloud Object Gateway from the MCG command-line interface.
- **Support of Multicloud Object Gateway data bucket APIs**
  The following table lists the Multicloud Object Gateway (MCG) data bucket APIs and their support levels.

Accessing the Multicloud Object Gateway from the terminal

Use this information to access the Multicloud Object Gateway from the terminal.

Before you begin

- A running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager.
  - For IBM Power, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
    ```

  - For IBM Z infrastructure, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
    ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.
  Note: Choose the correct Product Variant according to your architecture.

Procedure

Run the **describe** command to view information about the Multicloud Object Gateway (MCG) endpoint, including its access key (AWS_ACCESS_KEY_ID value) and secret access key (AWS_SECRET_ACCESS_KEY value).

```
oc describe noobaa -n openshift-storage
```

Example output, where **(1)** is the access key (AWS_ACCESS_KEY_ID value), **(2)** is the secret access key (AWS_SECRET_ACCESS_KEY value), and **(3)** is the MCG endpoint

```
Name:         noobaa
Namespace:    openshift-storage
Labels:       <none>
Annotations:  <none>
API Version:  noobaa.io/v1alpha1
Kind:         NooBaa
Metadata:
  Creation Timestamp:  2019-07-29T16:22:06Z
  Generation:          1
  Resource Version:    6718822
  Self Link:           /apis/noobaa.io/v1alpha1/namespaces/openshift-storage/noobaas/noobaa
  UID:                 019cfb4a-b21d-11e9-9a02-06c8de012f9e
Spec:
Status:
  Accounts:
    Admin:
      Secret Ref:
        Name:           noobaa-admin
        Namespace:      openshift-storage
  Actual Image:         noobaa/noobaa-core:4.0
  Observed Generation:  1
  Phase:                Ready
  Readme:

  Welcome to NooBaa!
  ------------------

  Welcome to NooBaa!
    ------------------
    NooBaa Core Version:
    NooBaa Operator Version:

    Lets get started:

    1. Connect to Management console:

       Read your mgmt console login information (email & password) from secret: "noobaa-admin".
```

```
        kubectl get secret noobaa-admin -n openshift-storage -o json | jq '.data|map_values(@base64d)'

      Open the management console service - take External IP/DNS or Node Port or use port forwarding:

        kubectl port-forward -n openshift-storage service/noobaa-mgmt 11443:443 &
        open https://localhost:11443

    2. Test S3 client:

      kubectl port-forward -n openshift-storage service/s3 10443:443 &how
(1)
      NOOBAA_ACCESS_KEY=$(kubectl get secret noobaa-admin -n openshift-storage -o json | jq -r
'.data.AWS_ACCESS_KEY_ID|@base64d')
(2)
      NOOBAA_SECRET_KEY=$(kubectl get secret noobaa-admin -n openshift-storage -o json | jq -r
'.data.AWS_SECRET_ACCESS_KEY|@base64d')
      alias s3='AWS_ACCESS_KEY_ID=$NOOBAA_ACCESS_KEY AWS_SECRET_ACCESS_KEY=$NOOBAA_SECRET_KEY aws --endpoint
https://localhost:10443 --no-verify-ssl s3'
      s3 ls


    Services:
      Service Mgmt:
        External DNS:
          https://noobaa-mgmt-openshift-storage.apps.mycluster-cluster.qe.rh-ocs.com
          https://a3406079515be11eaa3b70683061451e-1194613580.us-east-2.elb.amazonaws.com:443
        Internal DNS:
          https://noobaa-mgmt.openshift-storage.svc:443
        Internal IP:
          https://172.30.235.12:443
        Node Ports:
          https://10.0.142.103:31385
        Pod Ports:
          https://10.131.0.19:8443
      serviceS3:
        External DNS: (3)
          https://s3-openshift-storage.apps.mycluster-cluster.qe.rh-ocs.com
          https://a340f4e1315be11eaa3b70683061451e-943168195.us-east-2.elb.amazonaws.com:443
        Internal DNS:
          https://s3.openshift-storage.svc:443
        Internal IP:
          https://172.30.86.41:443
        Node Ports:
          https://10.0.142.103:31011
        Pod Ports:
          https://10.131.0.19:6443
```

Note: The output from the **oc describe noobaa** command lists the internal and external DNS names that are available. When using the internal DNS, the traffic is free. The external DNS uses Load Balancing to process the traffic, and therefore has a cost per hour.

Accessing the Multicloud Object Gateway from the MCG command-line interface

Use this information to access the Multicloud Object Gateway from the MCG command-line interface.

Before you begin

- A running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms

  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager.
  - For IBM Power, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
    ```

  - For IBM Z infrastructure, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
    ```

Procedure

Run the **status** command to access the endpoint, access key, and secret access key.

```
odf noobaa status -n openshift-storage
```

Example output, where **(1)** is the endpoint, **(2)** is the access key, and **(3)** is secret access key.

```
INFO[0000] Namespace: openshift-storage
INFO[0000]
INFO[0000] CRD Status:
INFO[0003] ✅ Exists: CustomResourceDefinition "noobaas.noobaa.io"
INFO[0003] ✅ Exists: CustomResourceDefinition "backingstores.noobaa.io"
INFO[0003] ✅ Exists: CustomResourceDefinition "bucketclasses.noobaa.io"
INFO[0004] ✅ Exists: CustomResourceDefinition "objectbucketclaims.objectbucket.io"
INFO[0004] ✅ Exists: CustomResourceDefinition "objectbuckets.objectbucket.io"
```

```
INFO[0004]
INFO[0004] Operator Status:
INFO[0004] ✅ Exists: Namespace "openshift-storage"
INFO[0004] ✅ Exists: ServiceAccount "noobaa"
INFO[0005] ✅ Exists: Role "ocs-operator.v0.0.271-6g45f"
INFO[0005] ✅ Exists: RoleBinding "ocs-operator.v0.0.271-6g45f-noobaa-f9vpj"
INFO[0006] ✅ Exists: ClusterRole "ocs-operator.v0.0.271-fjhgh"
INFO[0006] ✅ Exists: ClusterRoleBinding "ocs-operator.v0.0.271-fjhgh-noobaa-pdxn5"
INFO[0006] ✅ Exists: Deployment "noobaa-operator"
INFO[0006]
INFO[0006] System Status:
INFO[0007] ✅ Exists: NooBaa "noobaa"
INFO[0007] ✅ Exists: StatefulSet "noobaa-core"
INFO[0007] ✅ Exists: Service "noobaa-mgmt"
INFO[0008] ✅ Exists: Service "s3"
INFO[0008] ✅ Exists: Secret "noobaa-server"
INFO[0008] ✅ Exists: Secret "noobaa-operator"
INFO[0008] ✅ Exists: Secret "noobaa-admin"
INFO[0009] ✅ Exists: StorageClass "openshift-storage.noobaa.io"
INFO[0009] ✅ Exists: BucketClass "noobaa-default-bucket-class"
INFO[0009] ✅ (Optional) Exists: BackingStore "noobaa-default-backing-store"
INFO[0010] ✅ (Optional) Exists: CredentialsRequest "noobaa-cloud-creds"
INFO[0010] ✅ (Optional) Exists: PrometheusRule "noobaa-prometheus-rules"
INFO[0010] ✅ (Optional) Exists: ServiceMonitor "noobaa-service-monitor"
INFO[0011] ✅ (Optional) Exists: Route "noobaa-mgmt"
INFO[0011] ✅ (Optional) Exists: Route "s3"
INFO[0011] ✅ Exists: PersistentVolumeClaim "db-noobaa-core-0"
INFO[0011] ✅ System Phase is "Ready"
INFO[0011] ✅ Exists:  "noobaa-admin"

#------------------#
#- Mgmt Addresses -#
#------------------#

ExternalDNS : [https://noobaa-mgmt-openshift-storage.apps.mycluster-cluster.qe.rh-ocs.com
https://a3406079515be11eaa3b70683061451e-1194613580.us-east-2.elb.amazonaws.com:443]
ExternalIP  : []
NodePorts   : [https://10.0.142.103:31385]
InternalDNS : [https://noobaa-mgmt.openshift-storage.svc:443]
InternalIP  : [https://172.30.235.12:443]
PodPorts    : [https://10.131.0.19:8443]

#--------------------#
#- Mgmt Credentials -#
#--------------------#

email    : admin@noobaa.io
password : HKLbH1rSuVU0I/souIkSiA==

#----------------#
#- S3 Addresses -#
#----------------#

(1)
ExternalDNS : [https://s3-openshift-storage.apps.mycluster-cluster.qe.rh-ocs.com https://a340f4e1315be11eaa3b70683061451e-
943168195.us-east-2.elb.amazonaws.com:443]
ExternalIP  : []
NodePorts   : [https://10.0.142.103:31011]
InternalDNS : [https://s3.openshift-storage.svc:443]
InternalIP  : [https://172.30.86.41:443]
PodPorts    : [https://10.131.0.19:6443]

#------------------#
#- S3 Credentials -#
#------------------#

(2)
AWS_ACCESS_KEY_ID     : jVmAsu9FsvRHYmfjTiHV
(3)
AWS_SECRET_ACCESS_KEY : E//420VNedJfATvVSmDz6FMtsSAzuBv6z180PT5c

#------------------#
#- Backing Stores -#
#------------------#

NAME                          TYPE     TARGET-BUCKET                                              PHASE   AGE
noobaa-default-backing-store  aws-s3   noobaa-backing-store-15dc896d-7fe0-4bed-9349-5942211b93c9   Ready   141h35m32s

#------------------#
#- Bucket Classes -#
#------------------#

NAME                          PLACEMENT                                                          PHASE   AGE
noobaa-default-bucket-class   {Tiers:[{Placement: BackingStores:[noobaa-default-backing-store]}]}   Ready   141h35m33s

#-----------------#
#- Bucket Claims -#
#-----------------#

No OBC's found.
```

Results

You now have the relevant endpoint, access key, and secret access key in order to connect to your applications.

For example:

If AWS S3 CLI is the application, the following command lists the buckets in Fusion Data Foundation:

```
AWS_ACCESS_KEY_ID=<AWS_ACCESS_KEY_ID>
AWS_SECRET_ACCESS_KEY=<AWS_SECRET_ACCESS_KEY>
aws --endpoint <ENDPOINT> --no-verify-ssl s3 ls
```

Support of Multicloud Object Gateway data bucket APIs

The following table lists the Multicloud Object Gateway (MCG) data bucket APIs and their support levels.

Data buckets	Support	
List buckets	Supported	
Delete bucket	Supported	Replication configuration is part of MCG bucket class configuration
Create bucket	Supported	A different set of canned ACLs
Post bucket	Not supported	
Put bucket	Partially supported	Replication configuration is part of MCG bucket class configuration
Bucket lifecycle	Partially supported	Object expiration only
Policy (Buckets, Objects)	Partially supported	Bucket policies are supported
Bucket Website	Supported	
Bucket ACLs (Get, Put)	Supported	A different set of canned ACLs
Bucket Location	Partialy	Returns a default value only
Bucket Notification	Not supported	
Bucket Object Versions	Supoorted	
Get Bucket Info (HEAD)	Supported	
Bucket Request Payment	Partially supported	Returns the bucket owner
Put Object	Supported	
Delete Object	Supported	
Get Object	Supported	
Object ACLs (Get, Put)	Supported	
Get Object Info (HEAD)	Supported	
POST Object	Supported	
Copy Object	Supported	
Multipart Uploads	Supported	
Object Tagging	Supported	
Storage Class	Not supported	

Adding storage resources for hybrid or Multicloud

Understand how to add and edit storage resources for hybrid or Multicloud.

- **Creating a new backing store**
  Create a new backing store to add storage resources.
- **Overriding the default backing store**
  Follow this procedure to override the default NooBaa backing store.
- **Adding storage resources for hybrid or Multicloud using the MCG command line interface**
  The Multicloud Object Gateway (MCG) simplifies the process of spanning data across the cloud provider and clusters. dd a backing storage that can be used by the MCG.
- **Creating an s3 compatible Multicloud Object Gateway backingstore**
  Create an S3 compatible MCG backing store for IBM Storage Ceph's RGW. When the RGW is deployed, Fusion Data Foundation operator creates an S3 compatible backingstore for MCG automatically.
- **Creating a new bucket class**
  Bucket class is a CRD representing a class of buckets that defines tiering policies and data placements for an Object Bucket Class (OBC). Use this procedure to create a bucket class in Fusion Data Foundation.
- **Editing a bucket class**
  Use this information to edit an existing bucket class.
- **Editing backing stores for bucket class**
  Use this procedure to edit an existing Multicloud Object Gateway (MCG) bucket class to change the underlying backing stores used in a bucket class.

Creating a new backing store

Create a new backing store to add storage resources.

Before you begin

Ensure you have Administrator access to Fusion Data Foundation.

Procedure

1. From the OpenShift Web Console, go to Storage > Data Foundation.
2. From the Backing Store tab, click Create Backing Store.
3. From the Create New Backing Store page:
   a. Enter a Backing Store Name.
   b. Select a Provider.
   c. Select a Region.
   d. Optional: Enter an Endpoint..
   e. Select a Secret from the drop-down list, or create your own secret. Optionally, you can Switch to Credentials view which lets you fill in the required secrets.
   For more information on creating an OCP secret, see Security > Authentication and authorization > Configuring identity providers > Creating the secret in Red Hat OpenShift Container Platform product documentation.

   Each backingstore requires a different secret. For more information on creating the secret for a particular backingstore, see Adding storage resources for hybrid or Multicloud using the MCG command line interface and follow the steps to add storage resources using a YAML.

   Note: This menu is relevant for all providers except Google Cloud and local PVC.
   f. Enter the Target bucket.
   The target bucket is a storage that is hosted on the remote cloud service. The target bucket enables you to create a connection that tells the MCG that it can use this bucket for the system.
4. Click Create Backing Store.

What to do next

Optionally, verify the creation of the backing store:

1. From the OpenShift Web Console, go to Storage > Data Foundation.
2. From the Backing Store tab to view all the backing stores.

Overriding the default backing store

Follow this procedure to override the default NooBaa backing store.

Before you begin

Download the Multicloud Object Gateway (MCG) command-line interface.

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
 yum install mcg
```

Note: Specify the appropriate architecture for enabling the repositories using the subscription manager.

- For IBM Power, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
  ```

- For IBM Z infrastructure, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
  ```

About this task

You can use the `manualDefaultBackingStore`flag to override the default NooBaa backing store and remove it if you do not want to use the default backing store configuration. This provides flexibility to customize your backing store configuration and tailor it to your specific needs. By leveraging this feature, you can further optimize your system and enhance its performance.

Procedure

1. Check if `noobaa-default-backing-store` is present:

   ```
   oc get backingstore
   NAME TYPE PHASE AGE
   noobaa-default-backing-store pv-pool Creating 102s
   ```

2. Patch the NooBaa CR to enable `manualDefaultBackingStore`:

   ```
   oc patch noobaa/noobaa --type json --patch='[{"op":"add","path":"/spec/manualDefaultBackingStore","value":true}]'
   ```

   Important: Use the Multicloud Object Gateway CLI to create a new backing store and update accounts.
3. Create a new default backing store to override the default backing store.
   For example:

   ```
   odf noobaa backingstore create pv-pool _NEW-DEFAULT-BACKING-STORE_ --num-volumes 1 --pv-size-gb 16
   ```

   a. Replace `NEW-DEFAULT-BACKING-STORE` with the name you want for your new default backing store.
4. Update the admin account to use the new default backing store as its default resource:

   ```
   odf noobaa account update admin@noobaa.io --new_default_resource=_NEW-DEFAULT-BACKING-STORE_
   ```

   a. Replace `NEW-DEFAULT-BACKING-STORE` with the name of the backing store from the previous step.

Updating the default resource for admin accounts ensures that the new configuration is used throughout your system.

5. Configure the default-bucketclass to use the new default backingstore:

```
oc patch Bucketclass noobaa-default-bucket-class -n openshift-storage --type=json --patch='[{"op": "replace", "path": "/spec/placementPolicy/tiers/0/backingStores/0", "value": "NEW-DEFAULT-BACKING-STORE"}]'
```

6. Optional: Delete the `noobaa-default-backing-store`.
   a. Delete all instances of and buckets associated with `noobaa-default-backing-store` and update the accounts using it as resource.
   b. Delete the `noobaa-default-backing-store`:

```
oc delete backingstore noobaa-default-backing-store -n openshift-storage | oc patch -n openshift-storage backingstore/noobaa-default-backing-store --type json --patch='[ { "op": "remove", "path": "/metadata/finalizers" } ]'
```

   You must enable the `manualDefaultBackingStore` flag before proceeding. Additionally, it is crucial to update all accounts that use the default resource and delete all instances of and buckets associated with the default backing store to ensure a smooth transition.

Adding storage resources for hybrid or Multicloud using the MCG command line interface

The Multicloud Object Gateway (MCG) simplifies the process of spanning data across the cloud provider and clusters. dd a backing storage that can be used by the MCG.

Depending on your deployment type, follow the appropriate procedure to add storage resources by creating a backing storage.

For VMware deployments, skip this section and go to Creating an s3 compatible Multicloud Object Gateway backingstore.

- **Creating an AWS-backed backingstore**
  Create an AWS-backed backing store using the MCG command-line interface or a YAML file to add storage resources.
- **Creating an AWS-STS-backed backingstore**
  Creating an AWS STS-backed backingstore involves creating an AWS role with temporary credentials, installing the operator in an AWS STS cluster, and creating the backingstore using short-lived credentials.
- **Creating an IBM COS-backed backingstore**
  Create an IBM COS-backed backing store using the MCG command-line interface or a YAML file to add storage resources.
- **Creating an Azure-backed backingstore**
  Create an Azure-backed backing store using the MCG command-line interface or a YAML file to add storage resources.
- **Creating a GCP-backed backingstore**
  Create a GCP-backed backing store using the MCG command-line interface or a YAML file to add storage resources.
- **Creating a local Persistent Volume-backed backingstore**
  Create a local Persistent Volume-backed backing store using the MCG command-line interface or a YAML file to add storage resources.

Creating an AWS-backed backingstore

Create an AWS-backed backing store using the MCG command-line interface or a YAML file to add storage resources.

Before you begin, ensure the following:

- Fusion Data Foundation Platform is running.
- Download the MCG command-line interface for easier management.

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
```

```
yum install mcg
```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager. In case of IBM Z infrastructure, use the following command:

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

Create an AWS-backed backingstore using MCG command-line interface

From the MCG command-line interface, run the following command:

```
odf noobaa backingstore create aws-s3 <backingstore_name> --access-key=<AWS ACCESS KEY> --secret-key=<AWS SECRET ACCESS KEY> --target-bucket <bucket-name> -n openshift-storage
```

backingstore_name
: The name of the backingstore.

AWS ACCESS KEY and *AWS SECRET ACCESS KEY*
: The AWS access key ID and secret access key you created for this purpose.

bucket-name
: The existing AWS bucket name. This argument indicates to the MCG which bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.

Example output:

```
INFO[0001] ✅ Exists: NooBaa "noobaa"
INFO[0002] ✅ Created: BackingStore "aws-resource"
INFO[0002] ✅ Created: Secret "backing-store-secret-aws-resource"
```

Create an AWS-backed backingstore using a YAML file

1. Create a secret with the credentials:

```
apiVersion: v1
kind: Secret
metadata:
  name: <backingstore-secret-name>
  namespace: openshift-storage
type: Opaque
data:
  AWS_ACCESS_KEY_ID: <AWS ACCESS KEY ID ENCODED IN BASE64>
  AWS_SECRET_ACCESS_KEY: <AWS SECRET ACCESS KEY ENCODED IN BASE64>
```

AWS ACCESS KEY and *AWS SECRET ACCESS KEY*
: Supply and encode your own AWS access key ID and secret access key using Base64, and use the results for ***AWS ACCESS KEY ID ENCODED IN BASE64*** and ***AWS SECRET ACCESS KEY ENCODED IN BASE64***.

backingstore-secret-name
: The name of the backingstore secret created in Creating a new backing store.

2. Apply the following YAML for a specific backing store:

```
apiVersion: noobaa.io/v1alpha1
kind: BackingStore
metadata:
  finalizers:
  - noobaa.io/finalizer
  labels:
    app: noobaa
  name: bs
  namespace: openshift-storage
spec:
  awsS3:
    secret:
      name: <backingstore-secret-name>
      namespace: openshift-storage
    targetBucket: <bucket-name>
  type: aws-s3
```

bucket-name
: The existing AWS bucket name.

backingstore-secret-name
: The name of the backingstore secret created in the previous step.

Creating an AWS-STS-backed backingstore

Creating an AWS STS-backed backingstore involves creating an AWS role with temporary credentials, installing the operator in an AWS STS cluster, and creating the backingstore using short-lived credentials.

Amazon Web Services Security Token Service (AWS STS) is an AWS feature and it is a way to authenticate using short-lived credentials. Creating an AWS-STS-backed backingstore involves the following:

- Creating an AWS role using a script, which helps to get the temporary security credentials for the role session
- Installing {product-name-short} operator in AWS STS OpenShift cluster
- Creating backingstore in AWS STS OpenShift cluster

- **Creating an AWS role using a script**
  You need to create a role and pass the role Amazon resource name (ARN) while installing the {product-name-short} operator.
- **Installing OpenShift Data Foundation operator in AWS STS OpenShift cluster**
  Install the OpenShift Data Foundation operator in an AWS STS OpenShift cluster by annotating the subscription with the ARN and applying the required OIDC-based credentials.
- **Creating a new AWS STS backingstore**
  Create an AWS STS backingstore for the Multicloud Object Gateway using short-lived credentials obtained through OpenID Connect (OIDC) authentication.

Creating an AWS role using a script

You need to create a role and pass the role Amazon resource name (ARN) while installing the {product-name-short} operator.

Before you begin

Configure Red Hat OpenShift Container Platform cluster with AWS STS. For more information, see Configuring an AWS cluster to use short-term credentials.

Procedure

Create an AWS role using a script that matches OpenID Connect (OIDC) configuration for Multicloud Object Gateway (MCG) on OpenShift Data Foundation. The following example shows the details that are required to create the role:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::123456789123:oidc-provider/mybucket-oidc.s3.us-east-2.amazonaws.com"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "mybucket-oidc.s3.us-east-2.amazonaws.com:sub": [
                        "system:serviceaccount:openshift-storage:noobaa",
                        "system:serviceaccount:openshift-storage:noobaa-endpoint"
                    ]
                }
            }
        }
    ]
}
```

where

`123456789123`
: Is the AWS account ID

`mybucket`
: Is the bucket name (using public bucket configuration)

`us-east-2`
: Is the AWS region

`openshift-storage`
: Is the namespace name

Sample script

```
#!/bin/bash
set -x

# This is a sample script to help you deploy MCG on AWS STS cluster.
# This script shows how to create role-policy and then create the role in AWS.
# For more information see: https://docs.openshift.com/rosa/authentication/assuming-an-aws-iam-role-for-a-service-account.html

# WARNING: This is a sample script. You need to adjust the variables based on your requirement.

# Variables :
# user variables - REPLACE these variables with your values:
ROLE_NAME="<role-name>" # role name that you pick in your AWS account
NAMESPACE="<namespace>" # namespace name where MCG is running. For OpenShift Data Foundation, it is openshift-storage.

# MCG variables
SERVICE_ACCOUNT_NAME_1="<service-account-name-1>" # The service account name of statefulset core and deployment operator (MCG
SERVICE_ACCOUNT_NAME_2="<service-account-name-2>" # The service account name of deployment endpoint (MCG endpoint)

# AWS variables
# Ensure these values are not empty (AWS_ACCOUNT_ID, OIDC_PROVIDER)
# AWS_ACCOUNT_ID is your AWS account number
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
# If you want to create the role before using the cluster, replace this field too.
# The OIDC provider is in the structure:
# 1) <OIDC-bucket>.s3.<aws-region>.amazonaws.com. for OIDC bucket configurations are in an S3 public bucket
# 2) `<characters>.cloudfront.net` for OIDC bucket configurations in an S3 private bucket with a public CloudFront distributio
OIDC_PROVIDER=$(oc get authentication cluster -ojson | jq -r .spec.serviceAccountIssuer | sed -e "s/^https:\/\///")
# the permission (S3 full access)
POLICY_ARN_STRINGS="arn:aws:iam::aws:policy/AmazonS3FullAccess"

# Creating the role (with AWS command line interface)

read -r -d '' TRUST_RELATIONSHIP <<EOF
{
 "Version": "2012-10-17",
 "Statement": [
   {
         "Effect": "Allow",
         "Principal": {
         "Federated": "arn:aws:iam::${AWS_ACCOUNT_ID}:oidc-provider/${OIDC_PROVIDER}"
         },
         "Action": "sts:AssumeRoleWithWebIdentity",
         "Condition": {
         "StringEquals": {
         "${OIDC_PROVIDER}:sub": [
         "system:serviceaccount:${NAMESPACE}:${SERVICE_ACCOUNT_NAME_1}",
         "system:serviceaccount:${NAMESPACE}:${SERVICE_ACCOUNT_NAME_2}"
         ]
         }
         }
   }
 ]
```

```
}
EOF

echo "${TRUST_RELATIONSHIP}" > trust.json

aws iam create-role --role-name "$ROLE_NAME" --assume-role-policy-document file://trust.json --description "role for demo"

while IFS= read -r POLICY_ARN; do
   echo -n "Attaching $POLICY_ARN ... "
   aws iam attach-role-policy \
        --role-name "$ROLE_NAME" \
        --policy-arn "${POLICY_ARN}"
   echo "ok."
done <<< "$POLICY_ARN_STRINGS"
```

Installing OpenShift Data Foundation operator in AWS STS OpenShift cluster

Install the OpenShift Data Foundation operator in an AWS STS OpenShift cluster by annotating the subscription with the ARN and applying the required OIDC-based credentials.

Before you begin

- Configure OpenShift Container Platform cluster with AWS STS. For more information, see Configuring an AWS cluster to use short-term credentials.
- Create an AWS role using a script that matches OpenID Connect (OIDC) configuration. For more information, see Creating an AWS role using a script.

Procedure

Install Fusion Data Foundation Operator from the Operator Hub.

- During the installation add the role ARN in the **ARN Details** field.
- Ensure that the **Update approval** field is set to **Manual**.

Creating a new AWS STS backingstore

Create an AWS STS backingstore for the Multicloud Object Gateway using short-lived credentials obtained through OpenID Connect (OIDC) authentication.

Before you begin

- Configure Red Hat OpenShift Container Platform cluster with AWS STS. For more information, see Configuring an AWS cluster to use short-term credentials.
- Create an AWS role using a script that matches OpenID Connect (OIDC) configuration. For more information, see Creating an AWS role using a script.

Procedure

1. Install Multicloud Object Gateway (MCG).
   It is installed with the default backingstore by using the short-lived credentials.
2. After the MCG system is ready, you can create more backingstores of the type `aws-sts-s3` using the following MCG command line interface command:

   ```
   odf noobaa backingstore create aws-sts-s3 <backingstore-name> --aws-sts-arn=<aws-sts-role-arn> --region=<region> --target-bucket=<target-bucket>
   ```

   where

   backingstore-name
   : Name of the backingstore

   aws-sts-role-arn
   : The AWS STS role ARN that assumes the role

   region
   : The AWS bucket region

   target-bucket
   : The target bucket name on the cloud

Creating an IBM COS-backed backingstore

Create an IBM COS-backed backing store using the MCG command-line interface or a YAML file to add storage resources.

Before you begin, ensure the following:

- Fusion Data Foundation Platform is running.
- Download the MCG command-line interface for easier management.

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms

yum install mcg
```

Note: Specify the appropriate architecture for enabling the repositories using the subscription manager.

- For IBM Power, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
  ```

- For IBM Z infrastructure, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
  ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

Create an IBM COS-backed backingstore using MCG command-line interface

From the MCG command-line interface, run the following command:

```
odf noobaa backingstore create ibm-cos <backingstore_name> --access-key=<IBM ACCESS KEY> --secret-key=<IBM SECRET ACCESS KEY> --endpoint=<IBM COS ENDPOINT> --target-bucket <bucket-name> -n openshift-storage
```

backingstore_name
: The name of the backingstore.

IBM ACCESS KEY and *IBM SECRET ACCESS KEY*
: An IBM access key ID, secret access key and the appropriate regional endpoint that corresponds to the location of the existing IBM bucket.
To generate the preceding keys on IBM cloud, you must include HMAC credentials while creating the service credentials for your target bucket.

bucket-name
: The existing IBM bucket name. This argument indicates to the MCG which bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.
Example output:

```
INFO[0001] ✅ Exists: NooBaa "noobaa"
INFO[0002] ✅ Created: BackingStore "ibm-resource"
INFO[0002] ✅ Created: Secret "backing-store-secret-ibm-resource"
```

Create an IBM COS-backed backingstore using a YAML file

1. Create a secret with the credentials:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: <backingstore-secret-name>
     namespace: openshift-storage
   type: Opaque
   data:
     IBM_COS_ACCESS_KEY_ID: <IBM COS ACCESS KEY ID ENCODED IN BASE64>
     IBM_COS_SECRET_ACCESS_KEY: <IBM COS SECRET ACCESS KEY ENCODED IN BASE64>
   ```

   IBM ACCESS KEY and *IBM SECRET ACCESS KEY*
   : Provide and encode your own IBM COS access key ID and secret access key using Base64, and use the results in place of these attributes respectively.

   backingstore-secret-name
   : The name of the backingstore secret created in Creating a new backing store.

2. Apply the following YAML for a specific backing store:

   ```
   apiVersion: noobaa.io/v1alpha1
   kind: BackingStore
   metadata:
     finalizers:
     - noobaa.io/finalizer
     labels:
       app: noobaa
     name: bs
     namespace: openshift-storage
   spec:
     ibmCos:
       endpoint: <endpoint>
       secret:
         name: <backingstore-secret-name>
         namespace: openshift-storage
       targetBucket: <bucket-name>
     type: ibm-cos
   ```

   bucket-name
   : An existing IBM COS bucket name.

   endpoint
   : A regional endpoint that corresponds to the location of the existing IBM bucket name. This argument indicates to MCG about the endpoint to use for its backingstore, and subsequently, data storage and administration.

   backingstore-secret-name
   : The name of the backingstore secret created in the previous step.

Creating an Azure-backed backingstore

Create an Azure-backed backing store using the MCG command-line interface or a YAML file to add storage resources.

Before you begin, ensure the following:

- Fusion Data Foundation Platform is running.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms

  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager. In case of IBM Z infrastructure, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
  ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

Create an Azure-backed backingstore using MCG command-line interface

From the MCG command-line interface, run the following command:

```
odf noobaa backingstore create azure-blob <backingstore_name> --account-key=<AZURE ACCOUNT KEY> --account-name=<AZURE ACCOUNT NAME> --target-blob-container <blob container name> -n openshift-storage
```

backingstore_name
: The name of the backingstore.

AZURE ACCOUNT KEY and *AZURE ACCOUNT NAME*
: An AZURE account key and account name you created for this purpose.

blob container name
: An existing Azure blob container name. This argument indicates to MCG about the bucket to use as a target bucket for its backingstore, and subsequently, data storage and administration.

Example output:

```
INFO[0001] ✅ Exists: NooBaa "noobaa"
INFO[0002] ✅ Created: BackingStore "azure-resource"
INFO[0002] ✅ Created: Secret "backing-store-secret-azure-resource"
```

Create an Azure-backed backingstore using a YAML file

1. Create a secret with the credentials:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: <backingstore-secret-name>
   type: Opaque
   data:
     AccountName: <AZURE ACCOUNT NAME ENCODED IN BASE64>
     AccountKey: <AZURE ACCOUNT KEY ENCODED IN BASE64>
   ```

   AZURE ACCOUNT NAME ENCODED IN BASE64 and *AZURE ACCOUNT KEY ENCODED IN BASE64*
   : Supply and encode your own Azure Account Name and Account Key using Base64, and use the results in place of these attributes respectively.

   backingstore-secret-name
   : A unique name of backingstore secret.

2. Apply the following YAML for a specific backingstore:

   ```
   apiVersion: noobaa.io/v1alpha1
   kind: BackingStore
   metadata:
     finalizers:
     - noobaa.io/finalizer
     labels:
       app: noobaa
     name: bs
     namespace: openshift-storage
   spec:
     azureBlob:
       secret:
         name: <backingstore-secret-name>
         namespace: openshift-storage
       targetBlobContainer: <blob-container-name>
     type: azure-blob
   ```

   blob-container-name
   : An existing Azure blob container name. This argument indicates to the MCG about the bucket to use as a target bucket for its backingstore, and subsequently, data storage and administration.

   backingstore-secret-name

The name of the backingstore secret created in Creating a new backing store.

Creating a GCP-backed backingstore

Create a GCP-backed backing store using the MCG command-line interface or a YAML file to add storage resources.

Before you begin, ensure the following:

- Fusion Data Foundation Platform is running.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager. In case of IBM Z infrastructure, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
  ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

Create a GCP-backed backingstore using MCG command-line interface

From the MCG command-line interface, run the following command:

```
odf noobaa backingstore create google-cloud-storage <backingstore_name> --private-key-json-file=<PATH TO GCP PRIVATE KEY JSON FILE> --target-bucket <GCP bucket name> -n openshift-storage
```

backingstore_name
: Name of the backingstore.

<PATH TO GCP PRIVATE KEY JSON FILE>
: A path to your GCP private key created for this purpose.

GCP bucket name
: An existing GCP object storage bucket name. This argument tells the MCG which bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.
The output is similar to the following:

```
INFO[0001] ✅ Exists: NooBaa "noobaa"
INFO[0002] ✅ Created: BackingStore "google-gcp"
INFO[0002] ✅ Created: Secret "backing-store-google-cloud-storage-gcp"
```

Create a GCP-backed backingstore using a YAML file

1. Create a secret with the credentials:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: <backingstore-secret-name>
   type: Opaque
   data:
     GoogleServiceAccountPrivateKeyJson: <GCP PRIVATE KEY ENCODED IN BASE64>
   ```

   GCP PRIVATE KEY ENCODED IN BASE64
   : Provide and encode your own GCP service account private key using Base64, and use the results for this attribute.

   backingstore-secret-name
   : A unique name of the backingstore secret.

2. Apply the following YAML for a specific backing store:

   ```
   apiVersion: noobaa.io/v1alpha1
   kind: BackingStore
   metadata:
     finalizers:
     - noobaa.io/finalizer
     labels:
       app: noobaa
     name: bs
     namespace: openshift-storage
   spec:
     googleCloudStorage:
       secret:
         name: <backingstore-secret-name>
         namespace: openshift-storage
       targetBucket: <target bucket>
     type: google-cloud-storage
   ```

   target bucket
   : An existing Google storage bucket. This argument indicates to the MCG about the bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.

backingstore-secret-name
The name of the backingstore secret created in Creating a new backing store.

Creating a local Persistent Volume-backed backingstore

Create a local Persistent Volume-backed backing store using the MCG command-line interface or a YAML file to add storage resources.

Before you begin, ensure the following:

- Fusion Data Foundation Platform is running.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager.
  - For IBM Power, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
    ```

  - For IBM Z infrastructure, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
    ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

Create a local Persistent Volume-backed backingstore using MCG command-line interface

From the MCG command-line interface, run the following command:
Note: This command must be run from within the `openshift-storage` namespace.

```
odf noobaa -n openshift-storage backingstore create pv-pool <backingstore_name> --num-volumes <NUMBER OF VOLUMES>  --pv-size-gb <VOLUME SIZE> --request-cpu <CPU REQUEST> --request-memory <MEMORY REQUEST> --limit-cpu <CPU LIMIT> --limit-memory <MEMORY LIMIT> --storage-class <LOCAL STORAGE CLASS>
```

backingstore_name
The name of the backingstore.

NUMBER OF VOLUMES
The number of volumes you would like to create. Increasing the number of volumes scales up the storage.

VOLUME SIZE
Required size in GB of each volume.

CPU REQUEST
Guaranteed amount of CPU requested in CPU unit `m`.

MEMORY REQUEST
Guaranteed amount of memory requested.

CPU LIMIT
Maximum amount of CPU that can be consumed in CPU unit `m`.

MEMORY LIMIT
Maximum amount of memory that can be consumed.

LOCAL STORAGE CLASS
The local storage class name, recommended to use `ocs-storagecluster-ceph-rbd`.

Create a local Persistent Volume-backed backingstore using a YAML file

Apply the following YAML for a specific backing store:

```
apiVersion: noobaa.io/v1alpha1
kind: BackingStore
metadata:
  finalizers:
  - noobaa.io/finalizer
  labels:
    app: noobaa
  name: <backingstore_name>
  namespace: openshift-storage
spec:
   pvPool:
    numVolumes: <NUMBER OF VOLUMES>
    resources:
      requests:
        storage: <VOLUME SIZE>
        cpu: <CPU REQUEST>
        memory: <MEMORY REQUEST>
      limits:
        cpu: <CPU LIMIT>
```

```
        memory: <MEMORY LIMIT>
    storageClass: <LOCAL STORAGE CLASS>
  type: pv-pool
```

backingstore_name
: The name of the backingstore.

NUMBER OF VOLUMES
: The number of volumes you would like to create. increasing the number of volumes scales up the storage.

VOLUME SIZE
: Required size in GB of each volume.

CPU REQUEST
: Guaranteed amount of CPU requested in CPU unit `m`.

MEMORY REQUEST
: Guaranteed amount of memory requested.

CPU LIMIT
: Maximum amount of CPU that can be consumed in CPU unit `m`.

MEMORY LIMIT
: Maximum amount of memory that can be consumed.

LOCAL STORAGE CLASS
: The local storage class name, recommended to use `ocs-storagecluster-ceph-rbd`.

Example output:

```
INFO[0001] ✅ Exists: NooBaa "noobaa"
INFO[0002] ✅ Exists: BackingStore "local-mcg-storage"
```

Creating an s3 compatible Multicloud Object Gateway backingstore

Create an S3 compatible MCG backing store for IBM Storage Ceph's RGW. When the RGW is deployed, Fusion Data Foundation operator creates an S3 compatible backingstore for MCG automatically.

The Multicloud Object Gateway (MCG) can use any S3 compatible object storage as a backing store. For example, IBM Storage Ceph's RADOS Object Gateway (RGW).

Creating an s3 compatible Multicloud Object Gateway backingstore using the MCG command-line interface

- Run the following command:
  Note: This command must be run from within the `openshift-storage` namespace.

  ```
  odf noobaa backingstore create s3-compatible rgw-resource --access-key=<RGW ACCESS KEY> --secret-key=<RGW SECRET KEY> --target-bucket=<bucket-name> --endpoint=<RGW endpoint> -n openshift-storage
  ```

  Example output:

  ```
  INFO[0001] ✅ Exists: NooBaa "noobaa"
  INFO[0002] ✅ Created: BackingStore "rgw-resource"
  INFO[0002] ✅ Created: Secret "backing-store-secret-rgw-resource"
  ```

- To get the *RGW ACCESS KEY* and *RGW SECRET KEY*, run the following command using your RGW user secret name:

  ```
  oc get secret <RGW USER SECRET NAME> -o yaml -n openshift-storage
  ```

  Decode the access key ID and the access key from Base64 and keep them.

  RGW ACCESS KEY
  RGW SECRET KEY
  : Decoded data
  bucket-name
  : An existing RGW bucket name. This argument tells the MCG which bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.
  RGW endpoint
  : To get the *RGW endpoint*, see Accessing the RADOS Object Gateway S3 endpoint.

Creating an s3 compatible Multicloud Object Gateway backingstore using a YAML file

1. Create a `CephObjectStore` user. This also creates a secret containing the RGW credentials:

   ```
   apiVersion: ceph.rook.io/v1
   kind: CephObjectStoreUser
   metadata:
     name: <RGW-Username>
     namespace: openshift-storage
   spec:
     store: ocs-storagecluster-cephobjectstore
     displayName: "<Display-name>"
   ```

RGW-Username
: A unique RGW user name.

Display-name
: A display name

2. Apply the following YAML for an S3-Compatible backing store:

```
apiVersion: noobaa.io/v1alpha1
kind: BackingStore
metadata:
  finalizers:
  - noobaa.io/finalizer
  labels:
    app: noobaa
  name: <backingstore-name>
  namespace: openshift-storage
spec:
  s3Compatible:
    endpoint: <RGW endpoint>
    secret:
      name: <backingstore-secret-name>
      namespace: openshift-storage
    signatureVersion: v4
    targetBucket: <RGW-bucket-name>

  type: s3-compatible
```

backingstore-secret-name
: Name of the secret that was created with `CephObjectStore` in the previous step.

bucket-name
: An existing RGW bucket name. This argument tells the MCG which bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.

RGW endpoint
: To get the *RGW endpoint*, see Accessing the RADOS Object Gateway S3 endpoint.

Creating a new bucket class

Bucket class is a CRD representing a class of buckets that defines tiering policies and data placements for an Object Bucket Class (OBC). Use this procedure to create a bucket class in Fusion Data Foundation.

Procedure

1. From the OpenShift Web Console, go to Storage > Data Foundation.
2. From the Bucket Class tab click Create Bucket Class.
3. On the Create new Bucket Class page, perform the following:
   a. Select the bucket class type by choosing one of the following options:
      - Standard: data is consumed by a Multicloud Object Gateway (MCG), deduped, compressed and encrypted.
      - Namespace: data is stored on the NamespaceStores without performing de-duplication, compression or encryption.

      By default, Standard is selected.

   b. Enter a Bucket Class Name.
   c. Click Next.
   d. In Placement Policy, select Tier 1 - Policy Type and click Next.
      You can choose one of the options as per your requirements.
      - Spread allows spreading of the data across the chosen resources.
      - Mirror allows full duplication of the data across the chosen resources.

   e. To add another policy tier, click Add Tier.
   f. Select at least one Backing Store resource from the available list if you have selected Tier 1 - Policy Type as Spread and click Next.
      Alternatively, you can also Creating a new backing store.
      Note: You need to select at least two backing stores when you select Policy Type as Mirror in step 3.d.

What to do next

To verify the creation of the new bucket class:

1. In the OpenShift Web Console, click **Storage** → **Data Foundation**.
2. Click the **Bucket Class** tab and search the new Bucket Class.

Editing a bucket class

Use this information to edit an existing bucket class.

Before you begin

Ensure you have Administrator access to OpenShift Web Console.

About this task

Use the following procedure to edit the bucket class components through the YAML file by clicking the edit button on the OpenShift web console.

Procedure

1. From the OpenShift Web Console, go to Storage > Data Foundation.
2. From the Bucket Class tab, next to the Bucket class you want to edit, click Action Menu > Edit Bucket Class.
   You are redirected to the YAML file.
3. Make the required changes in this file and click Save.

Editing backing stores for bucket class

Use this procedure to edit an existing Multicloud Object Gateway (MCG) bucket class to change the underlying backing stores used in a bucket class.

Before you begin

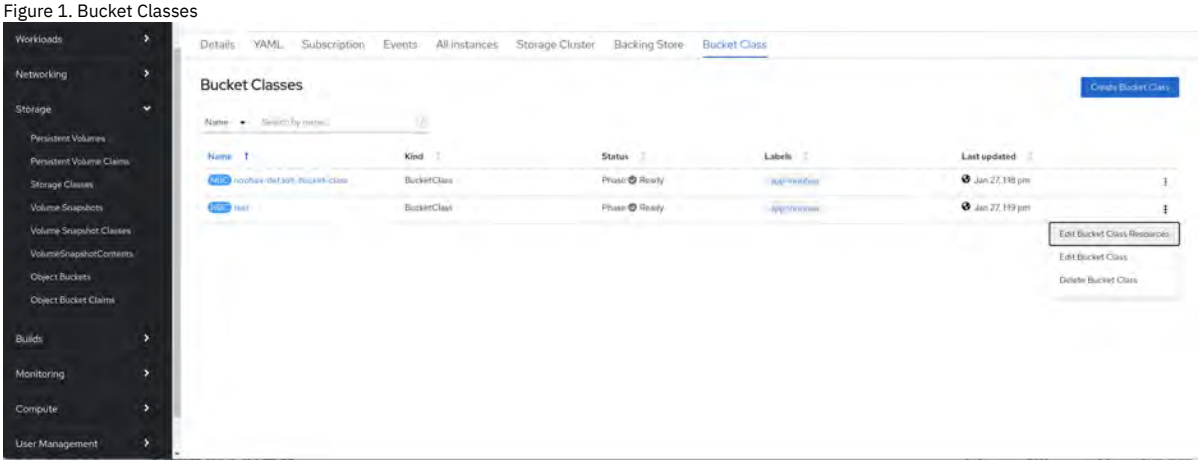
Ensure you have the following:

- Administrator access to OpenShift Web Console
- A bucket class
- Backing stores

Procedure

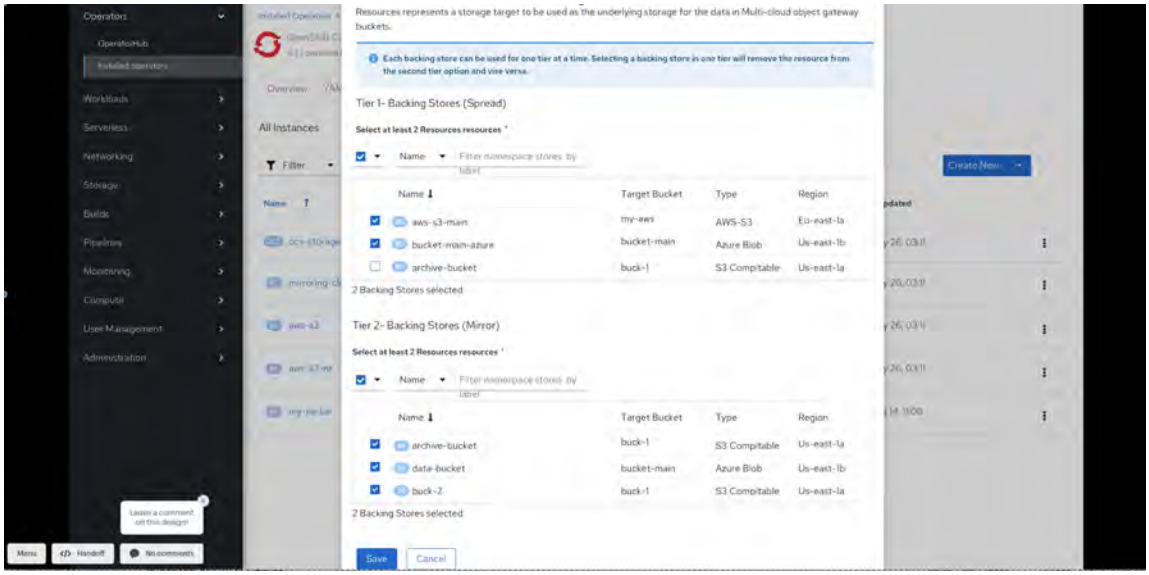
1. From the OpenShift Web Console, go to Storage > Data Foundation.
2. From the Bucket Class tab, next to the Bucket class you want to edit, click Action Menu > Edit Bucket Class Resources.

Figure 1. Bucket Classes



3. On the Edit Bucket Class Resources page, edit the bucket class resources.
   Edit the bucket class resources either by adding a backing store to the bucket class or by removing a backing store from the bucket class. You can also edit bucket class resources created with one or two tiers and different placement policies.
   - To add a backing store to the bucket class, select the name of the backing store.
   - To remove a backing store from the bucket class, clear the name of the backing store.

Figure 2. Editing backing stores



4. Click Save.

Creating and managing buckets using the object browser

The object browser in the OpenShift Web Console enables you to create, view, and manage buckets and objects across different storage endpoints, including Multicloud Object Gateway (MCG) and internal mode RADOS Gateway (RGW) object storage.

A storage endpoint selector is available under Storage > Object Storage > Buckets, allowing you to switch between the MCG and RGW. After selecting an endpoint, the object browser presents a unified interface for creating and managing buckets, uploading and viewing objects, configuring policies, and performing related operations.

- **Accessing the object browser**
  Learn how to access the object browser based on your user role: administrators or developers.
- **Creating and managing buckets**
  Create, view, and manage buckets for MCG and RGW storage endpoints using the object browser in the OpenShift Web Console.
- **Managing objects**
  Managing objects in the Multicloud Object Gateway includes uploading, downloading, and organizing S3-compatible objects stored across backing stores.
- **Creating and managing lifecycle rules**
  S3 bucket lifecycle rules automate object expiration and transition actions in the Multicloud Object Gateway to manage storage costs and data retention.
- **Creating and managing IAM user**
  Use this section to create and manage IAM users for object storage access.

Accessing the object browser

Learn how to access the object browser based on your user role: administrators or developers.

The access flow depends on your user role:

- **Administrator users**

  Administrators have direct access to the object browser without additional login steps.

- **Developer users**

  Developers must sign in before accessing the object browser as illustrated in Signing in to the object browser as a developer user.

- **Signing in to the object browser as a developer user**
  Sign in to the object browser with Kubernetes Secret credentials.

Signing in to the object browser as a developer user

Sign in to the object browser with Kubernetes Secret credentials.

Before you begin

- Kubernetes Secret containing S3 credentials
- Namespace and Secret name details

Procedure

1. Open the object browser in the OpenShift Web Console.
2. Select the storage endpoint:
   - MCG
   - RGW

   Developers must log in separately for each endpoint.
3. Select the Secret namespace and name where the S3 credentials are stored.
4. Optional: Select Stay signed in to remain logged in across browser tabs or sessions. Click Sign in.
   Note:
   - The UI does not store credentials in local or session storage but only a reference to the Kubernetes Secret is stored.
   - Access level depends on the credentials provided.

Creating and managing buckets

Create, view, and manage buckets for MCG and RGW storage endpoints using the object browser in the OpenShift Web Console.

- **Creating new buckets using the object browser**
  Create new buckets using the object browser through Object Bucket Claims or the S3 API.
- **Listing bucket details using the object browser**
  View bucket details in the object browser for MCG or RGW storage endpoints.
- **Deleting or emptying bucket using object browser**
  Delete or empty buckets by using the object browser in the OpenShift Web Console.

Creating new buckets using the object browser

Create new buckets using the object browser through Object Bucket Claims or the S3 API.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then click the Buckets tab.
3. Choose one of the following options to create the buckets:
   - **Create via Object Bucket Claim**
     Note: For RGW object bucket claims (OBCs), you must use the credentials associated with the OBC to access the newly created bucket. RGW administrator credentials cannot be used to access these buckets.
     - Select a namespace.
     - Enter the ObjectBucketClaim name.
     - Select the storage class.
     - Select the bucket class.
     - Select the Enable replication to obtain higher resiliency of the objects stored in the buckets.
     - Click Create using ObjectBucketClaim.
   - **Create via S3 API**
     - Enter a name for the bucket.
     - Add tags with different key and value criteria to tag your bucket.
     - Click Create.

Listing bucket details using the object browser

View bucket details in the object browser for MCG or RGW storage endpoints.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then click the Buckets tab.
3. Select the required bucket and then click the Details tab.

Deleting or emptying bucket using object browser

Delete or empty buckets by using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW and click the Buckets tab.
3. From the bucket's options menu, select Empty bucket.
4. Enter the bucket name to confirm.
5. Click Empty bucket.
6. After the bucket is empty, open the options menu again.
7. Select Delete bucket.
8. Enter the bucket name to confirm and click Delete bucket.

Managing objects

Managing objects in the Multicloud Object Gateway includes uploading, downloading, and organizing S3-compatible objects stored across backing stores.

- **Listing objects and object details using MCG object browser**
  View objects and object details using the object browser in the OpenShift Web Console.
- **Downloading or previewing objects using the object browser**
  Download or preview objects using the object browser in the OpenShift Web Console.
- **Generating the presigned URL to share the object**
  Generate the presigned URL to share an object using the object browser in the OpenShift Web Console.
- **Uploading objects using the object browser**
  Upload objects using the object browser in the OpenShift Web Console.
- **Creating folder using the object browser**
  Create a folder using the object browser in the OpenShift Web Console.
- **Deleting objects using the object browser**
  Delete objects using the object browser in the OpenShift Web Console.
- **Enabling or suspending object versioning in a bucket using the object browser**
  Enable or suspend objects versioning in a bucket using the object browser in the OpenShift Web Console.
- **Listing object versions using the object browser**
  View object versions using the object browser in the OpenShift Web Console.

Listing objects and object details using MCG object browser

View objects and object details using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, click Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW and click the Buckets tab.
3. Select the required bucket and click the Objects tab.
4. Select the object to open the details in the sidebar.

Downloading or previewing objects using the object browser

Download or preview objects using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.

- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, click Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the required bucket.
3. Click the Objects tab.
4. From the object's options menu, select Download or Preview.
   Note: Preview behavior depends on browser support. Unsupported file types that cannot be displayed in a separate tab are downloaded automatically.

Generating the presigned URL to share the object

Generate the presigned URL to share an object using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, click Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. Click the Objects tab.
4. From the object's options menu, select Share with presigned URL.
5. Select the expiration time to set the validity period of the presigned URL.
6. Click Create.
7. Copy the generated URL.

Uploading objects using the object browser

Upload objects using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. Click the Objects tab.
4. In the Add objects section, do one of the following:
   - Click Upload to add a folder.
   - Drag and drop individual files to upload specific files.
5. Click the uploaded object to view its Overview and Versions tab.

Creating folder using the object browser

Create a folder using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, click Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. Click the Objects tab.
4. Click Create folder.

5. Enter a folder name.
6. Click Create.
   Note: The folder persists only after at least one object is uploaded inside it.

Deleting objects using the object browser

Delete objects using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, click Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. Click the Objects tab.
4. To delete objects, do one of the following:
   - **Multiple objects**: Select multiple objects using the checkbox and then from the Actions menu, select Delete.
   - **Single object**: Use the object's Action menu and select Delete.

   Note: Folders cannot be deleted directly. Empty folders disappear automatically.

Enabling or suspending object versioning in a bucket using the object browser

Enable or suspend objects versioning in a bucket using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. Click the Properties tab.
4. Toggle Versioning to Enabled or Disabled.
5. Click Enable or Suspend to confirm depending on your versioning selection.
   Remember: Update lifecycle rules after enabling versioning.

Listing object versions using the object browser

View object versions using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. Click the Objects tab.
4. Toggle List of versions.
   - When List all versions is OFF:
     - The current version of the object excluding the delete markers is displayed.
     - Any Action menu options apply to the current version.
     - To view all versions of one object:
       - Click object and open the Versions tab from the sidebar.
   - When List all versions is ON:

- All object versions appear including deleted objects that have delete markers.
- Deleting a version is permanent.
- Removing a delete marker restores the most recent version. If no previous version exists, the object is permanently deleted.

Creating and managing lifecycle rules

S3 bucket lifecycle rules automate object expiration and transition actions in the Multicloud Object Gateway to manage storage costs and data retention.

- **Creating new lifecycle rules by using the object browser**
  Create lifecycle rules using the object browser in the OpenShift Web Console.
- **Editing lifecycle rules by using the object browser**
  Edit existing lifecycle rules using the object browser in the OpenShift Web Console.
- **Creating CORS rule**
  Configure cross-origin resource sharing (CORS) rules using the object browser in the OpenShift Web Console.
- **Editing CORS rule**
  Edit cross-origin resource sharing (CORS) rules using the object browser in the OpenShift Web Console.
- **Applying bucket policy to grant public or restricted access to objects**
  Apply or edit bucket policies using the object browser to control public or restricted access to objects.
- **Setting up public access limit to S3 resources**
  Configure public access restrictions for S3-compatible buckets by using the object browser.

Creating new lifecycle rules by using the object browser

Create lifecycle rules using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
   You can also create a new bucket using the steps from the Creating new buckets using the object browser.
3. Click the Management tab.
4. Click Create lifecycle rule.
5. Enter a name for the lifecycle rule.
6. Select a Rule scope:
   - Targeted (filters: prefix, tags, size)
   - Global (Bucket-wide)
7. Add one or more actions:
   - Delete an object after a specified time
     For a versioned bucket, the current version of the object is retained as a noncurrent version. For a nonversioned object, the object is deleted permanently.
   - Noncurrent versions of the objects
     This rule applies to only the versioned objects and deletes the older versions of the objects after they become noncurrent.
   - Incomplete multipart uploads
     This rule cleans up the abandoned uploads that were initiated but never completed after a specified time to prevent unnecessary storage costs. The object tags and size filter that were specified for the object with the targeted rule scope are not applicable for this rule and it deletes all multipart uploads from the bucket regardless of any filter.
   - Expired object delete markers
     This rule removes any unnecessary delete markers in versioned buckets that do not have any associated object versions, which might clutter bucket listings. The object tags and size filter that were specified for the object with the targeted rule scope are not applicable for this rule and it deletes all multipart uploads from the bucket regardless of any filter.

     Note:
     - The object tags and object size filters are not applicable for this rule. The expired object delete markers are removed from the bucket regardless of any filters that are configured.
     - This rule is not enabled when deleting an object after a specified time rule is selected.
8. Click Save.

Editing lifecycle rules by using the object browser

Edit existing lifecycle rules using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. In the bucket view, click the Management tab.
4. Locate the rule, then from the Action menu select Edit lifecycle rule.
5. Edit the required fields.
6. Click Save.

Creating CORS rule

Configure cross-origin resource sharing (CORS) rules using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. In the bucket view, click the Permission tab and then click the CORS tab.
4. Click Create CORS rule.
5. Provide a name for the rule.
6. Configure allowed origins*.
7. Select Allowed methods.
8. Configure allowed headers.
9. Enter the maximum age of preflight request in seconds.
10. Click Create.

Editing CORS rule

Edit cross-origin resource sharing (CORS) rules using the object browser in the OpenShift Web Console.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. In the bucket view, click the Permission tab and then click the CORS tab.
4. Select the CORS rule that you want to edit.
5. From the Actions menu, click Edit configuration.
6. Modify the required fields.
7. Click Save.

Applying bucket policy to grant public or restricted access to objects

Apply or edit bucket policies using the object browser to control public or restricted access to objects.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. In the bucket view, click the Permission tab.
4. In the Bucket policy tab, you can do one of the following:
   - When no policy exists, you can do one of the following:
     - Drag a file from another location or upload a file by using the Browse button.
     - Start from scratch or use a predefined policy configuration.

       Select Use a predefined policy configuration.

       - Create a new policy or choose a policy from Available policies.
       - Update the policy as required.
       - Click Apply policy.
       - Click Confirm to update the policy after reviewing the changes.
   - When a policy exists, use a predefined policy configuration by selecting the Edit bucket policy from the Action menu.
     - Select Use a predefined policy configuration.
     - Choose a policy from Available policies.
     - Update the policy as required.
     - Click Apply policy.
     - Click Confirm to update the policy after reviewing the changes.

   Note: After you apply the bucket policy configuration, it can take up to 120 seconds to take effect.

Setting up public access limit to S3 resources

Configure public access restrictions for S3-compatible buckets by using the object browser.

Before you begin

- **Administrators**: Administrator access to Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW, then open the bucket.
3. Select a bucket for which you need to set the public access limitation.
4. In the bucket view, click the Permission tab.
5. Click the Block public access tab.
6. Click Manage public access settings to set the limitations.
   Select one of the following options based on your requirement:
   - Block all public access.

     This blocks all forms of public access to the bucket and its objects.
   - Block public access to buckets and objects that are granted through new public bucket policies.

     This prevents new policies from making your data public.
   - Block public and cross-account access to buckets and objects through any public bucket policies.

     This blocks all public and cross-account access that is granted by any existing policies.
7. Click Save changes.

Creating and managing IAM user

Use this section to create and manage IAM users for object storage access.

Note the following considerations when creating and managing an IAM user:

- **Bucket policy principals and ARNs:** Bucket policies can reference either the account ID/ARN (for administrator or NooBaa CLI-created accounts) or the IAM user ARN. On upgraded clusters, bucket-policy principals previously defined using the email-based account format (account@noobaa.io) are automatically converted to the ARN format (arn:aws:iam::account_id:root).
- **S3 access requires a policy:** An IAM user's S3 requests return **AccessDenied** if no IAM policy is attached.
- **OBC account limitations:** Object Bucket Claim (OBC) accounts cannot create IAM users.
- **Default policy review:** IAM users created through the IAM console are automatically assigned a default policy that grants full access.

- **Creating an IAM user using the object browser**
  Learn how to create IAM users for object storage access using the object browser.
- **Generating additional access key for an IAM user**
  Learn how to generate an additional access key for an IAM user. You can generate one additional access key for the IAM user. IAM users can generate additional access key for themselves.

- **Activating or deactivating access key**
  You can generate one additional access key.
- **Deleting access key for an IAM user**
  Delete an access key for an IAM user.
- **Deleting an IAM user**
  Delete an IAM user from the object browser.

Creating an IAM user using the object browser

Learn how to create IAM users for object storage access using the object browser.

Before you begin

- **Administrators**: Administrator access to IBM Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW.
3. Click the IAM tab.
4. Click Create User.
5. Enter the required details in the Create User form.
   After the user is created, a window appears prompting you to download the secret key.
6. Click Download secret key, or copy the key and store it securely.
   Note:
   - When a new IAM user is created, an inline policy permitting S3 operations is generated automatically.
   - Access keys are also created automatically as part of the user creation workflow. No manual configuration is required.

Generating additional access key for an IAM user

Learn how to generate an additional access key for an IAM user. You can generate one additional access key for the IAM user. IAM users can generate additional access key for themselves.

Before you begin

- **Administrators**: Administrator access to IBM Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW.
3. Click the IAM tab.
4. Select the IAM user for whom you want to generate access key.
5. On the user details page, click the Access keys tab.
   The existing access key cards are displayed.
6. Click Generate another access key.
7. Optional: Enter a description for the new access key.
8. Click Generate.
   The newly generated access key card is displayed.

Results

The additional access key is generated and displayed in the Access keys tab.

Activating or deactivating access key

You can generate one additional access key.

Before you begin

- **Administrators**: Administrator access to IBM Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.

Note: Object Bucket Claim (OBC) based accounts do not have permissions to manage IAM users.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW.
3. Click the IAM tab.
4. Select the IAM user for whom you want to activate or deactivate access keys.
5. On the user details page, click the Access keys tab.
   The existing access key cards are displayed.
6. Select the Actions menu corresponding to the required Access key and select Activate/Deactivate Access key.
7. Click Activate or Deactivate depending on the existing status.

Deleting access key for an IAM user

Delete an access key for an IAM user.

Before you begin

- **Administrators**: Administrator access to IBM Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.
- An existing IAM user with at least one access key.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW.
3. Click the IAM tab.
4. Select the IAM user for whom you want to delete the access key.
5. On the user details page, click the Access keys tab.
   The list of access key cards are displayed.
6. Locate the access key you want to delete and open the Action menu on its card.
7. Select Delete access key.
8. You must deactivate the access key before deleting it. Click Deactivate.
9. To confirm deletion, type `delete` in the confirmation field.
10. Click Delete to permanently remove the access key.

Deleting an IAM user

Delete an IAM user from the object browser.

Before you begin

- **Administrators**: Administrator access to IBM Fusion Data Foundation.
- **Developers**: Logged in to the object browser using secret namespace and secret name. For more information see Accessing the object browser.
- Ensure that the selected storage endpoint (MCG or RGW) is deployed and its pods are healthy.
- All access keys and inline user policies associated with the user must be removed before deletion.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage.
2. From the Storage endpoint selector, choose MCG or RGW.
3. Click the IAM tab.
4. Select the IAM user you want to delete.
5. Click the Actions menu and select Delete user.
6. You must delete any existing access keys before deleting the user.
7. Type `delete` to confirm the user deletion.

Managing namespace buckets

Namespace buckets let you connect data repositories on different providers together, so you can interact with all of your data through a single unified view. Add the object bucket associated with each provider to the namespace bucket, and access your data through the namespace bucket to see all of your object buckets at once. This lets you write to your preferred storage provider while reading from multiple other storage providers, greatly reducing the cost of migrating to a new storage provider. Use this information to add namespace buckets using command-line interface, YAML, and user interface and for information about sharing and accessing legacy data.

Note: A namespace bucket can only be used if its write target is available and functional.

- **Amazon S3 API endpoints for objects in namespace buckets**
  Amazon S3 API endpoints supported for objects in namespace buckets list the operations that IBM Fusion Data Foundation supports when interacting with namespace bucket data.
- **Adding a namespace bucket using the Multicloud Object Gateway CLI and YAML**
  A namespace bucket can be added using either Multicloud Object Gateway command-line interface or YAML files.
- **Adding a namespace bucket using the OpenShift Container Platform user interface**
  You can add namespace buckets using the OpenShift Container Platform user interface.
- **Sharing legacy application data with cloud native application using S3 protocol**
  Many legacy applications use file systems to share data sets. You can access and share the legacy data in the file system by using the S3 operations.
- **Configuring or modifying the PublicAccessBlock configuration for S3 bucket**
  Learn how to restrict public access to S3 buckets, enhancing data security by configuring or modifying the PublicAccessBlock policy through `put-bucket-policy` S3 API.

Amazon S3 API endpoints for objects in namespace buckets

Amazon S3 API endpoints supported for objects in namespace buckets list the operations that IBM Fusion Data Foundation supports when interacting with namespace bucket data.

You can interact with objects in the namespace buckets using the Amazon Simple Storage Service (S3) API.

IBM Fusion Data Foundation supports the following namespace bucket operations:

- ListObjectVersions
- ListObjects
- PutObject
- CopyObject
- ListParts
- CreateMultipartUpload
- CompleteMultipartUpload
- UploadPart
- UploadPartCopy
- AbortMultipartUpload
- GetObjectAcl
- GetObject
- HeadObject
- DeleteObject
- DeleteObjects

See the Amazon S3 API reference documentation for the most up-to-date information about these operations and how to use them.

Related information

- Amazon S3 REST API Reference
- Amazon S3 CLI Reference

Adding a namespace bucket using the Multicloud Object Gateway CLI and YAML

A namespace bucket can be added using either Multicloud Object Gateway command-line interface or YAML files.

Depending on the type of your deployment and whether you want to use YAML or the Multicloud Object Gateway CLI, choose one of the following procedures to add a namespace bucket:

For more information about namespace buckets, see Managing namespace buckets.

- **Adding an AWS S3 namespace bucket using YAML**
  Add an AWS S3 namespace bucket using a YAML file.
- **Adding an IBM COS namespace bucket using YAML**
  Add an IBM COS namespace bucket using a YAML file.
- **Adding an AWS S3 namespace bucket using the Multicloud Object Gateway CLI**
  Add an AWS S3 namespace bucket using the Multicloud Object Gateway CLI.
- **Adding an IBM COS namespace bucket using the Multicloud Object Gateway CLI**
  Add an IBM COS namespace bucket using the Multicloud Object Gateway CLI.

Adding an AWS S3 namespace bucket using YAML

Add an AWS S3 namespace bucket using a YAML file.

Before you begin

- Install OpenShift Container Platform with Fusion Data Foundation operator.
- Ensure you have access to the Multicloud Object Gateway (MCG), see Accessing the Multicloud Object Gateway with your applications.

Procedure

1. Create a secret with the credentials:

```
apiVersion: v1
kind: Secret
metadata:
  name: <namespacestore-secret-name>
  type: Opaque
data:
  AWS_ACCESS_KEY_ID: <AWS ACCESS KEY ID ENCODED IN BASE64>
  AWS_SECRET_ACCESS_KEY: <AWS SECRET ACCESS KEY ENCODED IN BASE64>
```

namespacestore-secret-name
: Is a unique NamespaceStore name.

AWS ACCESS KEY ID ENCODED IN BASE64
AWS SECRET ACCESS KEY ENCODED IN BASE64
: You must provide and encode your own AWS access key ID and secret access key using **Base64**, and use the results in place of *AWS ACCESS KEY ID ENCODED IN BASE64* and *AWS SECRET ACCESS KEY ENCODED IN BASE64*.

2. Create a NamespaceStore resource using OpenShift custom resource definitions (CRDs).
A NamespaceStore represents underlying storage to be used as a **read** or **write** target for the data in the MCG namespace buckets.

   To create a NamespaceStore resource, apply the following YAML:

```
apiVersion: noobaa.io/v1alpha1
kind: NamespaceStore
metadata:
  finalizers:
  - noobaa.io/finalizer
  labels:
    app: noobaa
  name: <resource-name>
  namespace: openshift-storage
spec:
  awsS3:
    secret:
      name: <namespacestore-secret-name>
      namespace: <namespace-secret>
    targetBucket: <target-bucket>
  type: aws-s3
```

   resource-name
   : The name you want to give to the resource.

   namespacestore-secret-name
   : The secret created in the previous step.

   namespace-secret
   : The namespace where the secret can be found.

   target-bucket
   : The target bucket you created for the NamespaceStore.

3. Create a namespace bucket class that defines a namespace policy for the namespace buckets.
The namespace policy requires a type of either single or multi.
   - A namespace policy of type single requires the following configuration:

```
apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: noobaa
  name: <my-bucket-class>
  namespace: openshift-storage
spec:
  namespacePolicy:
    type:
    single:
      resource: <resource>
```

     my-bucket-class
     : The unique namespace bucket class name.

     resource
     : The name of a single NamespaceStore that defines the read and write target of the namespace bucket.

   - A namespace policy of type multi requires the following configuration:

```
apiVersion: noobaa.io/v1alpha1

kind: BucketClass
metadata:
  labels:
    app: noobaa
  name: my-bucket-class
  namespace: openshift-storage
spec:
  namespacePolicy:
    type: Multi
    multi:
```

```
      writeResource: <write-resource>
      readResources:
      - <read-resources>
      - <read-resources>
```

my-bucket-class
: A unique bucket class name.

write-resource
: The name of a single NamespaceStore that defines the **write** target of the namespace bucket.

read-resources
: A list of the names of the NamespaceStores that defines the **read** targets of the namespace bucket.

4. Create a bucket using an Object Bucket Class (OBC) resource.
   Use the bucket class defined in the earlier step using the following YAML:

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: <resource-name>
  namespace: openshift-storage
spec:
  generateBucketName: <my-bucket>
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: <my-bucket-class>
```

resource-name
: The name you want to give to the resource.

my-bucket
: The name you want to give to the bucket.

my-bucket-class
: The bucket class created in the previous step.

After the OBC is provisioned by the operator, a bucket is created in the MCG, and the operator creates a **Secret** and **ConfigMap** with the same name and in the same namespace as that of the OBC.

Adding an IBM COS namespace bucket using YAML

Add an IBM COS namespace bucket using a YAML file.

Before you begin

- Install OpenShift Container Platform with Fusion Data Foundation operator.
- Access to the Multicloud Object Gateway (MCG), see Accessing the Multicloud Object Gateway with your applications.

Procedure

1. Create a secret with the credentials:

```
apiVersion: v1
kind: Secret
metadata:
  name: <namespacestore-secret-name>
  type: Opaque
data:
  IBM_COS_ACCESS_KEY_ID: <IBM COS ACCESS KEY ID ENCODED IN BASE64>
  IBM_COS_SECRET_ACCESS_KEY: <IBM COS SECRET ACCESS KEY ENCODED IN BASE64>
```

namespacestore-secret-name
: A unique NamespaceStore name.

IBM COS ACCESS KEY ID ENCODED IN BASE64
IBM COS SECRET ACCESS KEY ENCODED IN BASE64
: You must provide and encode your own IBM COS access key ID and secret access key using **Base64**, and use the results in place of *IBM COS ACCESS KEY ID ENCODED IN BASE64* and *IBM COS SECRET ACCESS KEY ENCODED IN BASE64*

2. Create a NamespaceStore resource using OpenShift custom resource definitions (CRDs).
   A NamespaceStore represents underlying storage to be used as a **read** or **write** target for the data in the MCG namespace buckets.
   To create a NamespaceStore resource, apply the following YAML:

```
apiVersion: noobaa.io/v1alpha1
kind: NamespaceStore
metadata:
  finalizers:
  - noobaa.io/finalizer
  labels:
    app: noobaa
  name: bs
  namespace: openshift-storage
spec:
  s3Compatible:
```

```
    endpoint: <IBM COS ENDPOINT>
    secret:
      name: <namespacestore-secret-name>
      namespace: <namespace-secret>
    signatureVersion: v2
    targetBucket: <target-bucket>
  type: ibm-cos
```

IBM COS ENDPOINT
: The appropriate IBM COS endpoint.

namespacestore-secret-name
: The secret created in step 1.

namespace-secret
: The namespace where the secret can be found.

target-bucket
: The target bucket you created for the NamespaceStore.

3. Create a namespace bucket class that defines a namespace policy for the namespace buckets.
The namespace policy requires a type of either single or multi.
   - A namespace policy of type single requires the following configuration:

     ```
     apiVersion: noobaa.io/v1alpha1
     kind: BucketClass
     metadata:
       labels:
         app: noobaa
       name: <my-bucket-class>
       namespace: openshift-storage
     spec:
       namespacePolicy:
         type:
         single:
           resource: <resource>
     ```

     my-bucket-class
     : The unique namespace bucket class name.

     resource
     : The name of a single NamespaceStore that defines the read and write target of the namespace bucket.

   - A namespace policy of type multi requires the following configuration:

     ```
     apiVersion: noobaa.io/v1alpha1

     kind: BucketClass
     metadata:
       labels:
         app: noobaa
       name: my-bucket-class
       namespace: openshift-storage
     spec:
       namespacePolicy:
         type: Multi
         multi:
           writeResource: <write-resource>
           readResources:
           - <read-resources>
           - <read-resources>
     ```

     my-bucket-class
     : A unique bucket class name.

     write-resource
     : The name of a single NamespaceStore that defines the **write** target of the namespace bucket.

     read-resources
     : A list of the names of the NamespaceStores that defines the **read** targets of the namespace bucket.

4. Create a bucket using an Object Bucket Class (OBC) resource.
Use the bucket class defined in the earlier step using the following YAML:

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: <resource-name>
  namespace: openshift-storage
spec:
  generateBucketName: <my-bucket>
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: <my-bucket-class>
```

resource-name
: The name you want to give to the resource.

my-bucket
: The name you want to give to the bucket.

my-bucket-class

The bucket class created in the previous step.

After the OBC is provisioned by the operator, a bucket is created in the MCG, and the operator creates a **Secret** and **ConfigMap** with the same name and in the same namespace as that of the OBC.

Adding an AWS S3 namespace bucket using the Multicloud Object Gateway CLI

Add an AWS S3 namespace bucket using the Multicloud Object Gateway CLI.

Before you begin

- Install OpenShift Container Platform with Fusion Data Foundation operator.
- Ensure you have access to the Multicloud Object Gateway (MCG), see Accessing the Multicloud Object Gateway with your applications.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager. In case of IBM Z infrastructure, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
  ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

Procedure

1. In the MCG command-line interface, create a NamespaceStore resource.
   A NamespaceStore represents an underlying storage to be used as a **read** or **write** target for the data in MCG namespace buckets.

   ```
   odf noobaa namespacestore create aws-s3 <namespacestore> --access-key <AWS ACCESS KEY> --secret-key <AWS SECRET ACCESS KEY> --target-bucket <bucket-name> -n openshift-storage
   ```

   namespacestore
   The name of the NamespaceStore.

   AWS ACCESS KEY and *AWS SECRET ACCESS KEY*
   The AWS access key ID and secret access key you created for this purpose.

   bucket-name
   The existing AWS bucket name. This argument tells the MCG which bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.

2. Create a namespace bucket class that defines a namespace policy for the namespace buckets. The namespace policy can be either single or multi.
   - To create a namespace bucket class with a namespace policy of type single:

     ```
     odf noobaa bucketclass create namespace-bucketclass single <my-bucket-class> --resource <resource> -n openshift-storage
     ```

     resource-name
     The name you want to give the resource.
     my-bucket-class
     A unique bucket class name.
     resource
     A single namespace-store that defines the **read** and **write** target of the namespace bucket.

   - To create a namespace bucket class with a namespace policy of type multi:

     ```
     odf noobaa bucketclass create namespace-bucketclass multi <my-bucket-class> --write-resource <write-resource> --read-resources <read-resources> -n openshift-storage
     ```

     resource-name
     The name you want to give the resource.

     my-bucket-class
     A unique bucket class name.

     write-resource
     A single namespace-store that defines the **write** target of the namespace bucket.

     read-resources
     A list of namespace-stores separated by commas that defines the **read** targets of the namespace bucket.

3. Create a bucket using an Object Bucket Class (OBC) resource that uses the bucket class defined in the previous step.

   ```
   odf noobaa obc create my-bucket-claim -n openshift-storage --app-namespace my-app --bucketclass <custom-bucket-class>
   ```

   bucket-name
   A bucket name of your choice.

custom-bucket-class
The name of the bucket class created in the previous step.

After the OBC is provisioned by the operator, a bucket is created in the MCG, and the operator creates a **Secret** and a **ConfigMap** with the same name and in the same namespace as that of the OBC.

Adding an IBM COS namespace bucket using the Multicloud Object Gateway CLI

Add an IBM COS namespace bucket using the Multicloud Object Gateway CLI.

Before you begin

- Ensure you have a running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
```

```
yum install mcg
```

Note: Specify the appropriate architecture for enabling the repositories using the subscription manager.
  - For IBM Power, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
    ```

  - For IBM Z infrastructure, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
    ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

Procedure

1. In the MCG command-line interface, create a NamespaceStore resource.
   A NamespaceStore represents an underlying storage to be used as a **read** or **write** target for the data in the MCG namespace buckets.

   ```
   odf noobaa namespacestore create ibm-cos <namespacestore> --endpoint <IBM COS ENDPOINT> --access-key <IBM ACCESS KEY> --secret-key <IBM SECRET ACCESS KEY> --target-bucket <bucket-name> -n openshift-storage
   ```

   namespacestore
   The name of the NamespaceStore.

   IBM ACCESS KEY, *IBM SECRET ACCESS KEY*, **IBM COS ENDPOINT**
   An IBM access key ID, secret access key, and the appropriate regional endpoint that corresponds to the location of the existing IBM bucket.

   bucket-name
   An existing IBM bucket name. This argument tells the MCG which bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.

2. Create a namespace bucket class that defines a namespace policy for the namespace buckets. The namespace policy can be either single or multi.
   - To create a namespace bucket class with a namespace policy of type single:

     ```
     odf noobaa bucketclass create namespace-bucketclass single <my-bucket-class> --resource <resource> -n openshift-storage
     ```

     resource-name
     The name you want to give the resource.
     my-bucket-class
     A unique bucket class name.
     resource
     A single namespace-store that defines the **read** and **write** target of the namespace bucket.

   - To create a namespace bucket class with a namespace policy of type multi:

     ```
     odf noobaa bucketclass create namespace-bucketclass multi <my-bucket-class> --write-resource <write-resource> --read-resources <read-resources> -n openshift-storage
     ```

     resource-name
     The name you want to give the resource.

     my-bucket-class
     A unique bucket class name.

     write-resource
     A single namespace-store that defines the **write** target of the namespace bucket.

     read-resources
     A list of namespace-stores separated by commas that defines the **read** targets of the namespace bucket.

3. Create a bucket using an Object Bucket Class (OBC) resource that uses the bucket class defined in the previous step.

   ```
   odf noobaa obc create my-bucket-claim -n openshift-storage --app-namespace my-app --bucketclass <custom-bucket-class>
   ```

bucket-name
A bucket name of your choice.

custom-bucket-class
The name of the bucket class created in the previous step.

After the OBC is provisioned by the operator, a bucket is created in the MCG, and the operator creates a `Secret` and a `ConfigMap` with the same name and in the same namespace as that of the OBC.

Adding a namespace bucket using the OpenShift Container Platform user interface

You can add namespace buckets using the OpenShift Container Platform user interface.

Before you begin

- Install OpenShift Container Platform with Fusion Data Foundation operator.
- Access to the Multicloud Object Gateway (MCG).

About this task

For information about namespace buckets, see Managing namespace buckets.

Procedure

1. Log into the OpenShift Web Console and go to Storage > Object Storage > Namespace Store.
2. Create a `namespacestore` resource to be used in the namespace bucket.
    a. Click Create namespace store.
    b. Enter a namespacestore name.
    c. Choose a provider.
    d. Choose a region.
    e. Either select an existing secret, or click Switch to credentials to create a secret by entering a secret key and secret access key.
    f. Choose a target bucket.
    g. Click Create.
    h. Verify that the namespacestore is in the *Ready* state.
    i. Repeat these steps until you have the desired amount of resources.
3. From the Bucket Class tab, click Create a new Bucket Class.
    a. Select the Namespace.
    b. Enter a Bucket Class name.
    c. Optional: Add description.
    d. Click Next.
4. Choose a namespace policy type for your namespace bucket, and then click Next.
5. Select the target resources.
    - If your namespace policy type is Single, choose a read resource.
    - If your namespace policy type is Multi, choose read resources and a write resource.
    - If your namespace policy type is Cache, choose a Hub namespace store that defines the read and write target of the namespace bucket.
6. Click Next.
7. Review your new bucket class, and then click Create Bucketclass.
8. On the BucketClass page, verify that your newly created resource is in the *Created* phase.
9. Go to **Storage → Data Foundation**.
10. In the Status card, click Storage System and click the storage system link from the pop up that appears.
11. In the Object tab, go to Multicloud Object Gateway > Buckets > Namespace Buckets tab.
12. Click Create Namespace Bucket.
    a. From the Choose Name tab, specify a name for the namespace bucket and click Next.
    b. From the Set Placement tab:
        i. For Read Policy, select the check box for each namespace resource created in the earlier step that the namespace bucket should read data from.
        ii. If the namespace policy type you are using is Multi, then specify which namespace resource the namespace bucket should write data to within the Write Policy field.
    c. Click Next.
    d. Click Create.

What to do next

Verify that the namespace bucket is listed with a green check mark in the State column, the expected number of read resources, and the expected write resource name.

Sharing legacy application data with cloud native application using S3 protocol

Many legacy applications use file systems to share data sets. You can access and share the legacy data in the file system by using the S3 operations.

To share data:

- Export the pre-existing file system datasets, that is, RWX volume such as Ceph FileSystem (CephFS) or create a new file system datasets using the S3 protocol.
- Access file system datasets from both file system and S3 protocol.
- Configure S3 accounts and map them to the existing or a new file system unique identifiers (UIDs) and group identifiers (GIDs).

- **Creating a NamespaceStore to use a file system**
  Create a NamespaceStore to use a file system.
- **Creating accounts with NamespaceStore file system configuration**
  Create a Multicloud Object Gateway account with NamespaceStore file system configuration to enable file-system-backed object storage access by using S3 credentials.
- **Accessing legacy application data from the openshift-storage namespace**
  When using the Multicloud Object Gateway (MCG) NamespaceStore filesystem (NSFS) feature, you need to have the Persistent Volume Claim (PVC) where the data resides in the `openshift-storage` namespace. In almost all cases, the data you need to access is not in the `openshift-storage` namespace, but in the namespace that the legacy application uses. A PVC is used in order to access the data.

Creating a NamespaceStore to use a file system

Create a NamespaceStore to use a file system.

Before you begin

Ensure you have the following:

- OpenShift Container Platform with Fusion Data Foundation operator installed.
- Access to the Multicloud Object Gateway (MCG).

Procedure

1. Log into the OpenShift Web Console and go to Storage > Object Storage.
2. Go to the NamespaceStore tab to create NamespaceStore resources to be used in the namespace bucket.
3. Click Create namespacestore.
4. Enter a name for the NamespaceStore.
5. Choose Filesystem as the provider.
6. Choose the Persistent volume claim.
7. Enter a folder name.
   If the folder name exists, then that folder is used to create the NamespaceStore or else a folder with that name is created.
8. Click Create.
9. Verify the NamespaceStore is in the *Ready* state.

Creating accounts with NamespaceStore file system configuration

Create a Multicloud Object Gateway account with NamespaceStore file system configuration to enable file-system-backed object storage access by using S3 credentials.

Before you begin

Download the Multicloud Object Gateway (MCG) command-line interface:

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
```

```
yum install mcg
```

About this task

You can either create a new account with `NamespaceStore` file system configuration or convert an existing normal account into a `NamespaceStore` file system account by editing the YAML.
Note: You cannot remove a `NamespaceStore` file system configuration from an account.

Procedure

Create a new account with `NamespaceStore` file system configuration by using the MCG command-line interface.

```
odf noobaa account create <noobaa-account-name> [flags]
```

For example:

```
odf noobaa account create testaccount --nsfs_account_config --gid 10001 --uid 10001 -default_resource fs_namespacestore
```

`allow_bucket_create`	Indicates whether the account is allowed to create new buckets. Supported values are `true` or `false`. Default value is `true`.
`default_resource`	The NamespaceStore resource on which the new buckets are created when using the S3 CreateBucket operation. The NamespaceStore must be backed by a rwx (ReadWriteMany) persistent volume claim (PVC).
`force_md5_etag`	Enables md5 etag calculation for the user account.
`new_buckets_`	The file system path where directories corresponding to new buckets are created. The path is inside the file system of NamespaceStore file system

`path`	PVCs where new directories are created to act as the file system mapping of newly created object bucket classes.
`nsfs_account_config`	A mandatory field that indicates if the account is used for NamespaceStore file system.
`nsfs_only`	Indicates whether the account is used only for NamespaceStore file system or not. Supported values are true or `false`. Default value is `false`. If it is set to 'true', it limits you from accessing other types of buckets.
`uid`	The user ID of the file system to which the MCG account is mapped and it is used to access and manage data on the file system.
`gid`	The group ID of the file system to which the MCG account is mapped and it is used to access and manage data on the file system.

The MCG system sends a response with the account configuration and its S3 credentials:

```
NooBaaAccount spec:
allow_bucket_creation: true
Allowed_buckets:
  full_permission: true
  permission_list: []
default_resource: noobaa-default-namespace-store
Nsfs_account_config:
  gid: 10001
  new_buckets_path: /
  nsfs_only: true
  uid: 10001
INFO[0006] ✅ Exists: Secret "noobaa-account-testaccount"
Connection info:
  AWS_ACCESS_KEY_ID       : <aws-access-key-id>
  AWS_SECRET_ACCESS_KEY   : <aws-secret-access-key>
```

What to do next

You can list all the custom resource definition (CRD) based accounts by using the following command:

```
odf noobaa account list
```

```
NAME          ALLOWED_BUCKETS   DEFAULT_RESOURCE               PHASE   AGE
testaccount   [*]               noobaa-default-backing-store   Ready   1m17s
```

If you are interested in a particular account, you can read its custom resource definition (CRD) directly by the account name:

```
oc get noobaaaccount/testaccount -o yaml
spec:
  allow_bucket_creation: true
  allowed_buckets:
    full_permission: true
    permission_list: []
  default_resource: noobaa-default-namespace-store
  nsfs_account_config:
    gid: 10001
    new_buckets_path: /
    nsfs_only: true
    uid: 10001
```

Accessing legacy application data from the openshift-storage namespace

When using the Multicloud Object Gateway (MCG) NamespaceStore filesystem (NSFS) feature, you need to have the Persistent Volume Claim (PVC) where the data resides in the `openshift-storage` namespace. In almost all cases, the data you need to access is not in the `openshift-storage` namespace, but in the namespace that the legacy application uses. A PVC is used in order to access the data.

About this task

To access data stored in another namespace, you need to create a PVC in the `openshift-storage` namespace that points to the same CephFS volume that the legacy application uses.

Procedure

1. Display the application namespace with `scc`, where *<application_namespace>* is the name of the application namespace.

   ```
   oc get ns <application_namespace> -o yaml | grep scc
   ```

   For example:

   ```
   oc get ns testnamespace -o yaml | grep scc
   ```

   ```
   openshift.io/sa.scc.mcs: s0:c26,c5
   openshift.io/sa.scc.supplemental-groups: 1000660000/10000
   openshift.io/sa.scc.uid-range: 1000660000/10000
   ```

2. Navigate into the application namespace:.

   ```
   oc project <application_namespace>
   ```

   For example:

   ```
   oc project testnamespace
   ```

3. Ensure that a ReadWriteMany (RWX) PVC is mounted on the pod that you want to consume from the noobaa S3 endpoint using the MCG NSFS feature:.

   ```
   oc get pvc
   ```

```
NAME                                                STATUS VOLUME
CAPACITY ACCESS MODES STORAGECLASS              AGE
cephfs-write-workload-generator-no-cache-pv-claim  Bound  pvc-aa58fb91-c3d2-475b-bbee-68452a613e1a
10Gi     RWX          ocs-storagecluster-cephfs 12s

oc get pod


NAME                                                READY   STATUS              RESTARTS   AGE
cephfs-write-workload-generator-no-cache-1-cv892    1/1     Running             0          11s
```

4. Check the mount point of the Persistent Volume (PV) inside your pod.
   a. Get the volume name of the PV from the pod, where *<pod_name>* is the name of the pod.

      ```
      oc get pods <pod_name> -o jsonpath='{.spec.volumes[]}'
      ```

      For example:

      ```
      oc get pods cephfs-write-workload-generator-no-cache-1-cv892 -o jsonpath='{.spec.volumes[]}'

      {"name":"app-persistent-storage","persistentVolumeClaim":{"claimName":"cephfs-write-workload-generator-no-cache-pv-
      claim"}}
      ```

      In this example, the name of the volume for the PVC is **cephfs-write-workload-generator-no-cache-pv-claim**.

   b. List all the mounts in the pod, and check for the mount point of the volume that were identified in step 4.a.

      ```
      oc get pods <pod_name> -o jsonpath='{.spec.containers[].volumeMounts}'
      ```

      For example:

      ```
      oc get pods cephfs-write-workload-generator-no-cache-1-cv892 -o jsonpath='{.spec.containers[].volumeMounts}'


      [{"mountPath":"/mnt/pv","name":"app-persistent-storage"},
      {"mountPath":"/var/run/secrets/kubernetes.io/serviceaccount","name":"kube-api-access-8tnc5","readOnly":true}]
      ```

5. Confirm the mount point of the RWX PV in your pod, where *<mount_path>* is the path to the mount point that was identified in step 4.b.

   ```
   oc exec -it <pod_name> -- df <mount_path>
   ```

   For example:

   ```
   oc exec -it cephfs-write-workload-generator-no-cache-1-cv892 -- df /mnt/pv

   main
   Filesystem
   1K-blocks Used Available  Use%  Mounted on
   172.30.202.87:6789,172.30.120.254:6789,172.30.77.247:6789:/volumes/csi/csi-vol-cc416d9e-dbf3-11ec-b286-
   0a580a810213/edcfe4d5-bdcb-4b8e-8824-8a03ad94d67c
   10485760  0    10485760   0%    /mnt/pv
   ```

6. Ensure that the UID and SELinux labels are the same as the ones that the legacy namespace uses:.

   ```
   oc exec -it <pod_name> -- ls -latrZ <mount_path>
   ```

   For example:

   ```
   oc exec -it cephfs-write-workload-generator-no-cache-1-cv892 -- ls -latrZ /mnt/pv/

   total 567
   drwxrwxrwx. 3 root       root system_u:object_r:container_file_t:s0:c26,c5      2 May 25 06:35 .
   -rw-r--r--. 1 1000660000 root system_u:object_r:container_file_t:s0:c26,c5 580138 May 25 06:35 fs_write_cephfs-write-
   workload-generator-no-cache-1-cv892-data.log
   drwxrwxrwx. 3 root       root system_u:object_r:container_file_t:s0:c26,c5     30 May 25 06:35 ..
   ```

7. Get the information of the legacy application RWX PV that you want to make accessible from the **openshift-storage** namespace, where *<pv_name>* is the name of the PV.

   ```
   oc get pv | grep <pv_name>
   ```

   For example:

   ```
   oc get pv | grep pvc-aa58fb91-c3d2-475b-bbee-68452a613e1a

   pvc-aa58fb91-c3d2-475b-bbee-68452a613e1a   10Gi       RWX            Delete           Bound    testnamespace/cephfs-write-
   workload-generator-no-cache-pv-claim   ocs-storagecluster-cephfs              47s
   ```

8. Ensure that the PVC from the legacy application is accessible from the **openshift-storage** namespace so that one or more noobaa-endpoint pods can access the PVC.
   a. Find the values of the **subvolumePath** and **volumeHandle** from the **volumeAttributes**.
   You can get these values from the YAML description of the legacy application PV.

      ```
      oc get pv <pv_name> -o yaml
      ```

      For example:

      ```
      oc get pv pvc-aa58fb91-c3d2-475b-bbee-68452a613e1a -o yaml

      apiVersion: v1
      kind: PersistentVolume
      metadata:
        annotations:
          pv.kubernetes.io/provisioned-by: openshift-storage.cephfs.csi.ceph.com
        creationTimestamp: "2022-05-25T06:27:49Z"
      ```

```
  finalizers:
  - kubernetes.io/pv-protection
  name: pvc-aa58fb91-c3d2-475b-bbee-68452a613e1a
  resourceVersion: "177458"
  uid: 683fa87b-5192-4ccf-af2f-68c6bcf8f500
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 10Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: cephfs-write-workload-generator-no-cache-pv-claim
    namespace: testnamespace
    resourceVersion: "177453"
    uid: aa58fb91-c3d2-475b-bbee-68452a613e1a
  csi:
    controllerExpandSecretRef:
      name: rook-csi-cephfs-provisioner
      namespace: openshift-storage
    driver: openshift-storage.cephfs.csi.ceph.com
    nodeStageSecretRef:
      name: rook-csi-cephfs-node
      namespace: openshift-storage
    volumeAttributes:
      clusterID: openshift-storage
      fsName: ocs-storagecluster-cephfilesystem
      storage.kubernetes.io/csiProvisionerIdentity: 1653458225664-8081-openshift-storage.cephfs.csi.ceph.com
      subvolumeName: csi-vol-cc416d9e-dbf3-11ec-b286-0a580a810213
      subvolumePath: /volumes/csi/csi-vol-cc416d9e-dbf3-11ec-b286-0a580a810213/edcfe4d5-bdcb-4b8e-8824-8a03ad94d67c
                volumeHandle: 0001-0011-openshift-storage-0000000000000001-cc416d9e-dbf3-11ec-b286-0a580a810213
  persistentVolumeReclaimPolicy: Delete
  storageClassName: ocs-storagecluster-cephfs
  volumeMode: Filesystem
status:
  phase: Bound
```

b. Use the **subvolumePath** and **volumeHandle** values that you identified in step 8.a to create a new PV and PVC object in the **openshift-storage** namespace that points to the same CephFS volume as the legacy application PV.
Example YAML file:

```
cat << EOF >> pv-openshift-storage.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: cephfs-pv-legacy-openshift-storage
spec:
  storageClassName: ""
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 10Gi
                (1)
  csi:
    driver: openshift-storage.cephfs.csi.ceph.com
    nodeStageSecretRef:
      name: rook-csi-cephfs-node
      namespace: openshift-storage
    volumeAttributes:
    # Volume Attributes can be copied from the Source testnamespace PV
      "clusterID": "openshift-storage"
      "fsName": "ocs-storagecluster-cephfilesystem"
      "staticVolume": "true"
    # rootpath is the subvolumePath: you copied from the Source testnamespace PV
      "rootPath": /volumes/csi/csi-vol-cc416d9e-dbf3-11ec-b286-0a580a810213/edcfe4d5-bdcb-4b8e-8824-8a03ad94d67c
    volumeHandle: 0001-0011-openshift-storage-0000000000000001-cc416d9e-dbf3-11ec-b286-0a580a810213-clone
                (2)
  persistentVolumeReclaimPolicy: Retain
  volumeMode: Filesystem
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cephfs-pvc-legacy
  namespace: openshift-storage
spec:
  storageClassName: ""
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
                (3)
  volumeMode: Filesystem
  # volumeName should be same as PV name
  volumeName: cephfs-pv-legacy-openshift-storage
EOF
```

(1) The storage capacity of the PV that you are creating in the **openshift-storage** namespace must be the same as the original PV.

(2) The volume handle for the target PV that you create in **openshift-storage** needs to have a different handle than the original application PV, for example, add **-clone** at the end of the volume handle.

(3)
The storage capacity of the PVC that you are creating in the **openshift-storage** namespace must be the same as the original PVC.

c. Create the PV and PVC in the **openshift-storage** namespace using the YAML file specified in step 8.b, where *YAML_file* is the name of the YAML file.

```
oc create -f <YAML_file>
```

For example:

```
oc create -f pv-openshift-storage.yaml
```

```
persistentvolume/cephfs-pv-legacy-openshift-storage created
persistentvolumeclaim/cephfs-pvc-legacy created
```

d. Ensure that the PVC is available in the **openshift-storage** namespace:

```
oc get pvc -n openshift-storage
```

```
NAME                                 STATUS   VOLUME                                 CAPACITY   ACCESS MODES
STORAGECLASS                  AGE
cephfs-pvc-legacy                    Bound    cephfs-pv-legacy-openshift-storage     10Gi       RWX
14s
```

e. Navigate into the **openshift-storage** project:

```
oc project openshift-storage
```

```
Now using project "openshift-storage" on server "https://api.cluster-5f6ng.5f6ng.sandbox65.opentlc.com:6443".
```

f. Create the NSFS namespacestore:

```
odf noobaa namespacestore create nsfs <nsfs_namespacestore> --pvc-name='<cephfs_pvc_name>' --fs-backend='CEPH_FS'
```

nsfs_namespacestore
A name of the NSFS namespacestore.

cephfs_pvc_name
A name of the CephFS PVC in the **openshift-storage** namespace.

For example:

```
odf noobaa namespacestore create nsfs legacy-namespace --pvc-name='cephfs-pvc-legacy' --fs-backend='CEPH_FS'
```

g. Ensure that the noobaa-endpoint pod restarts and that it successfully mounts the PVC at the NSFS namespacestore, for example, **/nsfs/legacy-namespace** mountpoint, where *noobaa_endpoint_pod_name* is the name of the **noobaa-endpoint** pod.

```
oc exec -it <noobaa_endpoint_pod_name> -- df -h /nsfs/<nsfs_namespacestore>
```

For example:

```
oc exec -it noobaa-endpoint-5875f467f5-546c6 -- df -h /nsfs/legacy-namespace
```

```
Filesystem
Size  Used Avail Use% Mounted on
172.30.202.87:6789,172.30.120.254:6789,172.30.77.247:6789:/volumes/csi/csi-vol-cc416d9e-dbf3-11ec-b286-
0a580a810213/edcfe4d5-bdcb-4b8e-8824-8a03ad94d67c   10G     0   10G   0% /nsfs/legacy-namespace
```

h. Create an MCG user account:

```
odf noobaa account create <user_account> --allow_bucket_create=true --new_buckets_path='/' --nsfs_only=true --
nsfs_account_config=true --gid <gid_number> --uid <uid_number> --default_resource='legacy-namespace'
```

user_account
Specify the name of the MCG user account.

gid_number
Specify the GID number.

uid_number
Specify the UID number.

Important: Use the same **UID** and **GID** as that of the legacy application. You can find it from the previous output.

For example:

```
odf noobaa account create leguser --allow_bucket_create=true --new_buckets_path='/' --nsfs_only=true --
nsfs_account_config=true --gid 0 --uid 1000660000 --default_resource='legacy-namespace'
```

i. Create an MCG bucket.
   i. Create a dedicated folder for S3 inside the NSFS share on the CephFS PV and PVC of the legacy application pod:

   ```
   oc exec -it <pod_name> -- mkdir <mount_path>/nsfs
   ```

   For example:

   ```
   oc exec -it cephfs-write-workload-generator-no-cache-1-cv892 -- mkdir /mnt/pv/nsfs
   ```

   ii. Create the MCG bucket using the nsfs/ path:

   ```
   odf noobaa api bucket_api create_bucket '{
     "name": "<bucket_name>",
     "namespace":{
   ```

```
    "write_resource": { "resource": "<nsfs_namespacestore>", "path": "nsfs/" },
    "read_resources": [ { "resource": "<nsfs_namespacestore>", "path": "nsfs/" }]
  }
}'
```

For example:

```
odf noobaa api bucket_api create_bucket '{
  "name": "legacy-bucket",
  "namespace":{
    "write_resource": { "resource": "legacy-namespace", "path": "nsfs/" },
    "read_resources": [ { "resource": "legacy-namespace", "path": "nsfs/" }]
  }
}'
```

j. Check the SELinux labels of the folders residing in the PVCs in the legacy application and `openshift-storage` namespaces:

```
oc exec -it <noobaa_endpoint_pod_name> -n openshift-storage -- ls -ltraZ /nsfs/<nsfs_namespacstore>
```

For example:

```
oc exec -it noobaa-endpoint-5875f467f5-546c6 -n openshift-storage -- ls -ltraZ /nsfs/legacy-namespace
```

```
total 567
drwxrwxrwx. 3 root       root system_u:object_r:container_file_t:s0:c0,c26      2 May 25 06:35 .
-rw-r--r--. 1 1000660000 root system_u:object_r:container_file_t:s0:c0,c26 580138 May 25 06:35 fs_write_cephfs-write-
workload-generator-no-cache-1-cv892-data.log
drwxrwxrwx. 3 root       root system_u:object_r:container_file_t:s0:c0,c26     30 May 25 06:35 ..
```

```
oc exec -it <pod_name> -- ls -latrZ <mount_path>
```

For example:

```
oc exec -it cephfs-write-workload-generator-no-cache-1-cv892 -- ls -latrZ /mnt/pv/
```

```
total 567
drwxrwxrwx. 3 root       root system_u:object_r:container_file_t:s0:c26,c5      2 May 25 06:35 .
-rw-r--r--. 1 1000660000 root system_u:object_r:container_file_t:s0:c26,c5 580138 May 25 06:35 fs_write_cephfs-write-
workload-generator-no-cache-1-cv892-data.log
drwxrwxrwx. 3 root       root system_u:object_r:container_file_t:s0:c26,c5     30 May 25 06:35 ..
```

In these examples, you can see that the SELinux labels are not the same which results in permission denied or access issues.

9. Ensure that the legacy application and `openshift-storage` pods use the same SELinux labels on the files.
You can do this in one of the following ways:
   - Changing the default SELinux label on the legacy application project to match the one in the openshift-storage project
   - Modifying the SELinux label only for the deployment config that has the pod which mounts the legacy application PVC
10. Delete the NSFS namespacestore.
    a. Delete the MCG bucket:

    ```
    odf noobaa bucket delete <bucket_name>
    ```

    For example:

    ```
    odf noobaa bucket delete legacy-bucket
    ```

    b. Delete the MCG user account.

    ```
    odf noobaa account delete <user_account>
    ```

    For example:

    ```
    odf noobaa account delete leguser
    ```

    c. Delete the NSFS namespacestore.

    ```
    odf noobaa namespacestore delete <nsfs_namespacestore>
    ```

    For example:

    ```
    odf noobaa namespacestore delete legacy-namespace
    ```

11. Delete the PV and PVC.
Important: Before you delete the PV and PVC, ensure that the PV has a retain policy configured.

```
oc delete pv <cephfs_pv_name>
```

```
oc delete pvc <cephfs_pvc_name>
```

cephfs_pv_name
: Specify the CephFS PV name of the legacy application.

cephfs_pvc_name
: Specify the CephFS PVC name of the legacy application.

For example:

```
oc delete pv cephfs-pv-legacy-openshift-storage
```

For example:

```
oc delete pvc cephfs-pvc-legacy
```

- **Changing the default SELinux label on the legacy application project to match the one in the openshift-storage project**
  Ensure that the legacy application and `openshift-storage` pods use the same SELinux labels on the files, by changing the default SELinux label on the legacy application project.
- **Modifying the SELinux label only for the deployment config that has the pod which mounts the legacy application PVC**
  Ensure that the legacy application and `openshift-storage` pods use the same SELinux labels on the files, by modifying the SELinux label on the deployment config that has the pod which mounts the legacy application.

Changing the default SELinux label on the legacy application project to match the one in the openshift-storage project

Ensure that the legacy application and `openshift-storage` pods use the same SELinux labels on the files, by changing the default SELinux label on the legacy application project.

Procedure

1. Display the current `openshift-storage` namespace with `sa.scc.mcs`:

   ```
   oc get ns openshift-storage -o yaml | grep sa.scc.mcs
   openshift.io/sa.scc.mcs: s0:c26,c0
   ```

2. Edit the legacy application namespace, and modify the `sa.scc.mcs`.
   Use the value from the `sa.scc.mcs` of the `openshift-storage` namespace:

   ```
   oc edit ns <appplication_namespace>
   ```

   For example:

   ```
   oc edit ns testnamespace
   ```

   ```
   oc get ns <application_namespace> -o yaml | grep sa.scc.mcs
   ```

   For example:

   ```
   oc get ns testnamespace -o yaml | grep sa.scc.mcs
   openshift.io/sa.scc.mcs: s0:c26,c0
   ```

3. Restart the legacy application pod.
   A relabel of all the files take place and now the SELinux labels match with the `openshift-storage` deployment.

Modifying the SELinux label only for the deployment config that has the pod which mounts the legacy application PVC

Ensure that the legacy application and `openshift-storage` pods use the same SELinux labels on the files, by modifying the SELinux label on the deployment config that has the pod which mounts the legacy application.

Procedure

1. Create a new `scc` with the `MustRunAs` and `seLinuxOptions` options, with the Multi Category Security (MCS) that the `openshift-storage` project uses.
   Example YAML file:

   ```
   cat << EOF >> scc.yaml

   allowHostDirVolumePlugin: false
   allowHostIPC: false
   allowHostNetwork: false
   allowHostPID: false
   allowHostPorts: false
   allowPrivilegeEscalation: true
   allowPrivilegedContainer: false
   allowedCapabilities: null
   apiVersion: security.openshift.io/v1
   defaultAddCapabilities: null
   fsGroup:
     type: MustRunAs
   groups:
   - system:authenticated
   kind: SecurityContextConstraints
   metadata:
     annotations:
     name: restricted-pvselinux
   priority: null
   readOnlyRootFilesystem: false
   requiredDropCapabilities:
   - KILL
   - MKNOD
   - SETUID
   - SETGID
   runAsUser:
     type: MustRunAsRange
   seLinuxContext:
   ```

```
  seLinuxOptions:
    level: s0:c26,c0
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users: []
volumes:
- configMap
- downwardAPI
- emptyDir
- persistentVolumeClaim
- projected
- secret
EOF

oc create -f scc.yaml
```

2. Create a service account for the deployment and add it to the newly created **scc**.
   a. Create a service account, where *<service_account_name>* is the name of the service account.

      ```
      oc create serviceaccount <service_account_name>
      ```

      For example:

      ```
      oc create serviceaccount testnamespacesa
      ```

   b. Add the service account to the newly created **scc**:

      ```
      oc adm policy add-scc-to-user restricted-pvselinux -z <service_account_name>
      ```

      For example:

      ```
      oc adm policy add-scc-to-user restricted-pvselinux -z testname
      ```

3. Patch the legacy application deployment so that it uses the newly created service account.
   With this change, you can specify the SELinux label in the deployment.

   ```
   oc patch dc/<pod_name> '{"spec":{"template":{"spec":{"serviceAccountName": "<service_account_name>"}}}}'
   ```

   For example:

   ```
   oc patch dc/cephfs-write-workload-generator-no-cache --patch '{"spec":{"template":{"spec":{"serviceAccountName":
   "testnamespacesa"}}}}'
   ```

4. Edit the deployment to specify the security context to use at the SELinux label in the deployment configuration:

   ```
   oc edit dc <pod_name> -n <application_namespace>
   ```

   Add the following lines:

   ```
   spec:
    template:
       metadata:
         securityContext:
           seLinuxOptions:
             Level: <security_context_value>
   ```

   security_context_value
   : You can find this value when you run the command to create a dedicated folder for S3 inside the NSFS share on the CephFS PV and PVC of the legacy application pod.

   For example:

   ```
   oc edit dc cephfs-write-workload-generator-no-cache -n testnamespace
   ```

   ```
   spec:
    template:
       metadata:
         securityContext:
           seLinuxOptions:
             level: s0:c26,c0
   ```

5. Ensure that the security context to be used at the SELinux label in the deployment configuration is specified correctly.

   ```
   oc get dc <pod_name> -n <application_namespace> -o yaml | grep -A 2 securityContext
   ```

   For example:

   ```
   oc get dc cephfs-write-workload-generator-no-cache -n testnamespace -o yaml | grep -A 2 securityContext
   ```

   ```
   securityContext:
           seLinuxOptions:
             level: s0:c26,c0
   ```

   The legacy application is restarted and begins using the same SELinux labels as the **openshift-storage** namespace.

Configuring or modifying the PublicAccessBlock configuration for S3 bucket

Learn how to restrict public access to S3 buckets, enhancing data security by configuring or modifying the PublicAccessBlock policy through **put-bucket-policy** S3 API.

About this task

To apply PublicAccessBlock policy to the bucket, first apply a policy that grants public access, and then apply the PublicAccessBlock policy. The policy is applied through **`put-bucket-policy`** S3 API, which is similar to Amazon S3. For more information, see PutPublicAccessBlock in *Amazon S3 API Reference* Guide.

Important: The Multicloud Object Gateway (MCG) checks the PublicAccessBlock policy configuration for both the bucket that contains the object and the bucket owner's account during evaluating the PublicAccessBlock policy for a bucket or an object.
Restriction: Only bucket-wide policy is supported and not account-wide policy. Also, access control lists (ACLs) are not supported within the policy.

Procedure

1. Allow a general access policy.
   Command example:

```
{
  "Version": "2025-06-25",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::<bucket_name>/*",
        "arn:aws:s3:::<bucket_name>"
      ]
    }
  ]
}
```

2. Set the PublicAccessBlock policy.
   The default PublicAccessBlock policy configuration is as follows:

```
{
    {
        "BlockPublicPolicy": false,
        "RestrictPublicBuckets": false
    }
}
```

   To block public policies and restrict public buckets, set the value to **`true`** as follows:

```
{
    {
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    }
}
```

   The following configuration is not supported:

```
{
    {
        "BlockPublicAcls": false, <-- Not allowed even as false
        "IgnorePublicAcls": false, <-- Not allowed even as false
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    }
}
```

3. Apply the PublicAccessBlock policy by using the **`put-public-access-block`** command.
   Command example:

```
alias s3api=AWS_ACCESS_KEY_ID=<access_key> AWS_SECRET_ACCESS_KEY=<secret_key> aws s3api --no-verify-ssl --endpoint
<endpoint>  --no-verify put-public-access-block --bucket <bucket_name> --public-access-block-configuration
file://<configuration_file_path>. * *
```

Securing Multicloud Object Gateway

Ensure that the Multicloud Object Gateway is secure.

- **Changing the default account credentials to ensure better security in the Multicloud Object Gateway**
  Change and rotate your Multicloud Object Gateway (MCG) account credentials using the command-line interface to prevent issues with applications and to ensure better account security.
- **Enabling secured mode deployment for Multicloud Object Gateway**
  You can specify a range of IP addresses that should be allowed to reach the Multicloud Object Gateway (MCG) load balancer services to enable secure mode deployment. This helps to control the IP addresses that can access the MCG services.
- **Disabling routes that enable external access to Multicloud Object Gateway**
  Disable **`S3`**, **`STS`**, and **`noobaa-mgmt`** routes by updating the storagecluster CR.

Changing the default account credentials to ensure better security in the Multicloud Object Gateway

Change and rotate your Multicloud Object Gateway (MCG) account credentials using the command-line interface to prevent issues with applications and to ensure better account security.

Before changing the default account credentials, ensure you have the following:

- A running Fusion Data Foundation cluster.
- Download the Multicloud Object Gateway (MCG) command-line interface for easier management. For instructions, see Accessing the Multicloud Object Gateway with your applications.

- **Resetting the noobaa account password**
  Change and rotate your Multicloud Object Gateway (MCG) account credentials by resetting the noobaa account password.
- **Setting Multicloud Object Gateway account credentials using CLI command**
  Use this procedure to set the Multicloud Object Gateway (MCG) account credentials using the command-line interface (CLI) command. You can update and verify the MCG account credentials manually by using the MCG CLI command.
- **Regenerating the S3 credentials for the accounts**
  Change and rotate your Multicloud Object Gateway (MCG) account credentials by regenerating the S3 credentials for the accounts.
- **Regenerating the S3 credentials for the OBC**
  Change and rotate your Multicloud Object Gateway (MCG) account credentials by regenerating the S3 credentials for the OBC.

Resetting the noobaa account password

Change and rotate your Multicloud Object Gateway (MCG) account credentials by resetting the noobaa account password.

Procedure

To reset the noobaa account password, run the following command:

```
odf noobaa account passwd <noobaa_account_name> [options]

odf noobaa account passwd

FATA[0000] ✖ Missing expected arguments: <noobaa_account_name>

Options:
    --new-password='': New Password for authentication - the best practice is to omit this flag, in that
    case the CLI will prompt to prompt and read it securely from the terminal to avoid leaking secrets in t
     he shell history
    --old-password='': Old Password for authentication - the best practice is to omit this flag, in that
    case the CLI will prompt to prompt and read it securely from the terminal to avoid leaking secrets in
    the shell history
    --retype-new-password='': Retype new Password for authentication - the best practice is to omit
    this flag, in that case the CLI will prompt to prompt and read it securely from the terminal to avoid
    leaking secrets in the shell history


Usage:
    odf noobaa account passwd <noobaa-account-name> [flags] [options]

Use "odf noobaa options" for a list of global command-line options (applies to all commands).
```

Example:

```
odf noobaa account passwd admin@noobaa.io
```

Example output:

```
Enter old-password: [got 24 characters]
Enter new-password: [got 7 characters]
Enter retype-new-password: [got 7 characters]
INFO[0017] ✅ Exists: Secret "noobaa-admin"
INFO[0017] ✅ Exists: NooBaa "noobaa"
INFO[0017] ✅ Exists: Service "noobaa-mgmt"
INFO[0017] ✅ Exists: Secret "noobaa-operator"
INFO[0017] ✅ Exists: Secret "noobaa-admin"
INFO[0017] ✈️  RPC: account.reset_password() Request: {Email:admin@noobaa.io VerificationPassword:* Password:*}
WARN[0017] RPC: GetConnection creating connection to wss://localhost:58460/rpc/ 0xc000402ae0
INFO[0017] RPC: Connecting websocket (0xc000402ae0) &{RPC:0xc000501a40 Address:wss://localhost:58460/rpc/ State:init WS:<nil>
PendingRequests:map[] NextRequestID:0
Lock:{state:1 sema:0} ReconnectDelay:0s cancelPings:<nil>}
INFO[0017] RPC: Connected websocket (0xc000402ae0) &{RPC:0xc000501a40 Address:wss://localhost:58460/rpc/ State:init WS:<nil>
PendingRequests:map[] NextRequestID:0
Lock:{state:1 sema:0} ReconnectDelay:0s cancelPings:<nil>}
INFO[0020] ✅ RPC: account.reset_password() Response OK: took 2907.1ms
INFO[0020] ✅ Updated:  "noobaa-admin"
INFO[0020] ✅ Successfully reset the password for the account "admin@noobaa.io"
```

Important:
To access the admin account credentials run the **noobaa status** command from the terminal:

```
---------------------- Mgmt Credentials ----------------------

email    : admin@noobaa.io
password : ***
```

Setting Multicloud Object Gateway account credentials using CLI command

Use this procedure to set the Multicloud Object Gateway (MCG) account credentials using the command-line interface (CLI) command. You can update and verify the MCG account credentials manually by using the MCG CLI command.

Before you begin

Ensure that the following prerequisites are met:

- A running Fusion Data Foundation cluster.
- Download the MCG CLI binary from the customer portal and make it executable.

Note:

- Choose the correct product variant according to your architecture.
- Available platforms are Linux (x86_64), Windows, and Mac OS.

Procedure

1. Run the following command to update the MCG account credentials:

   ```
   odf noobaa account credentials <noobaa-account-name> [options]
   ```

   Example:

   ```
   odf noobaa account credentials admin@noobaa.io
   ```

   Example output:

   ```
   odf noobaa account credentials admin@noobaa.io
   Enter access-key: [got 20 characters]
   Enter secret-key: [got 40 characters]
   INFO[0026] ✖ Not Found: NooBaaAccount "admin@noobaa.io"
   INFO[0026] ✅ Exists: NooBaa "noobaa"
   INFO[0026] ✅ Exists: Service "noobaa-mgmt"
   INFO[0026] ✅ Exists: Secret "noobaa-operator"
   INFO[0026] ✅ Exists: Secret "noobaa-admin"
   INFO[0026] ✈  RPC: account.update_account_keys() Request: {Email:admin@noobaa.io AccessKeys:{AccessKey:* SecretKey:}}
   WARN[0026] RPC: GetConnection creating connection to wss://localhost:33495/rpc/ 0xc000cd9980 INFO[0026] RPC: Connecting
   websocket (0xc000cd9980) &{RPC:0xc0001655e0 Address:wss://localhost:33495/rpc/ State:init WS:<nil> PendingRequests:map[]
   NextRequestID:0 Lock:{state:1 sema:0} ReconnectDelay:0s cancelPings:<nil>} INFO[0026] RPC: Connected websocket
   (0xc000cd9980) &{RPC:0xc0001655e0 Address:wss://localhost:33495/rpc/ State:init WS:<nil> PendingRequests:map[]
   NextRequestID:0 Lock:{state:1 sema:0} ReconnectDelay:0s cancelPings:<nil>} INFO[0026] ✅ RPC:
   account.update_account_keys() Response OK: took 42.7ms INFO[0026] ✈ RPC: account.read_account() Request:
   {Email:admin@noobaa.io} INFO[0026] ✅ RPC: account.read_account() Response OK: took 2.0ms INFO[0026] ✅ Updated: "noobaa-
   admin" INFO[0026] ✅ Successfully updated s3 credentials for the account "admin@noobaa.io" INFO[0026] ✅ Exists: Secret
   "noobaa-admin" Connection info: AWS_ACCESS_KEY_ID :  AWS_SECRET_ACCESS_KEY : *
   ```

   Credential complexity requirements:

   Access key

   The account access key must be 20 characters in length, and it must contain only alphanumeric characters.

   Secret key

   The secret key must be 40 characters in length, and it must contain alphanumeric characters and "+" and "/".

   Example:

   ```
   odf noobaa account credentials my-account --access-key=ABCDEF1234567890ABCD --secret-
   key=ABCDE12345+FGHIJ67890/KLMNOPQRSTUV123456
   ```

2. Run the following command to verify the credentials:

   ```
   odf noobaa account status <noobaa-account-name> --show-secrets
   ```

   Remember: Each user must have a unique access key and secret key.

Regenerating the S3 credentials for the accounts

Change and rotate your Multicloud Object Gateway (MCG) account credentials by regenerating the S3 credentials for the accounts.

Procedure

1. Get the account name.
   - For listing the accounts, run the following command:

     ```
     odf noobaa account list
     ```

     Example output:

     ```
     NAME           ALLOWED_BUCKETS   DEFAULT_RESOURCE               PHASE   AGE
     account-test   [*]               noobaa-default-backing-store   Ready   14m17s
     test2          [first.bucket]    noobaa-default-backing-store   Ready   3m12s
     ```

   - Alternatively, run the **oc get noobaaaccount** command from the terminal:

```
oc get noobaaaccount
```

Example output:

```
NAME           PHASE   AGE
account-test   Ready   15m
test2          Ready   3m59s
```

2. To regenerate the noobaa account S3 credentials, run the following command:

```
odf noobaa account regenerate <noobaa_account_name> [options]
```

```
odf noobaa account regenerate
```

```
FATA[0000] ✖ Missing expected arguments: <noobaa-account-name>

Usage:
    odf noobaa account regenerate <noobaa-account-name> [flags] [options]

Use "odf noobaa options" for a list of global command-line options (applies to all commands).
```

3. After you run the **odf noobaa account regenerate** command it prompts the `This will invalidate all connections between S3 clients and NooBaa which are connected using the current credentials.` warning and asks for confirmation.
Example:

```
odf noobaa account regenerate account-test
```

Example output:

```
INFO[0000] You are about to regenerate an account's security credentials.
INFO[0000] This will invalidate all connections between S3 clients and NooBaa which are connected using the current
credentials.
INFO[0000] are you sure? y/n
```

After approving, the credentials are regenerated and eventually printed.

```
INFO[0015] ✓ Exists: Secret "noobaa-account-account-test"
Connection info:
AWS_ACCESS_KEY_ID       : ***
AWS_SECRET_ACCESS_KEY   : ***
```

Regenerating the S3 credentials for the OBC

Change and rotate your Multicloud Object Gateway (MCG) account credentials by regenerating the S3 credentials for the OBC.

Procedure

1. Get the OBC name.
   - Run the **odf noobaa obc list** command

     ```
     odf noobaa obc list
     ```

     Example output:

     ```
     NAMESPACE   NAME       BUCKET-NAME                                        STORAGE-CLASS       BUCKET-CLASS
     PHASE
     default     obc-test   obc-test-35800e50-8978-461f-b7e0-7793080e26ba   default.noobaa.io   noobaa-default-bucket-
     class   Bound
     ```

   - Run the **oc get obc** command from the terminal.

     ```
     oc get obc
     ```

     Example output:

     ```
     NAME       STORAGE-CLASS       PHASE   AGE
     obc-test   default.noobaa.io   Bound   38s
     ```

2. To regenerate the noobaa OBC S3 credentials, run the following command:

```
odf noobaa obc regenerate <bucket_claim_name> [options]
```

```
odf noobaa obc regenerate
FATA[0000] ✖ Missing expected arguments: <bucket-claim-name>

Usage:
   odf noobaa obc regenerate <bucket-claim-name> [flags] [options]

Use "odf noobaa options" for a list of global command-line options (applies to all commands).
```

3. After you run the `odf noobaa obc regenerate` command it prompts the `This will invalidate all connections between S3 clients and NooBaa which are connected using the current credentials.` warning and asks for confirmation.
Example:

```
odf noobaa obc regenerate obc-test
```

Example output:

```
INFO[0000] You are about to regenerate an OBC's security credentials.
INFO[0000] This will invalidate all connections between S3 clients and NooBaa which are connected using the current
```

```
credentials.
INFO[0000] are you sure? y/n
```

After approving, the credentials are regenerated and eventually printed.

```
INFO[0022] ✅ RPC: bucket.read_bucket() Response OK: took 95.4ms

ObjectBucketClaim info:
  Phase                 : Bound
  ObjectBucketClaim     : kubectl get -n default objectbucketclaim obc-test
  ConfigMap             : kubectl get -n default configmap obc-test
  Secret                : kubectl get -n default secret obc-test
  ObjectBucket          : kubectl get objectbucket obc-default-obc-test
  StorageClass          : kubectl get storageclass default.noobaa.io
  BucketClass           : kubectl get -n default bucketclass noobaa-default-bucket-class

Connection info:
 BUCKET_HOST            : s3.default.svc
 BUCKET_NAME            : obc-test-35800e50-8978-461f-b7e0-7793080e26ba
    BUCKET_PORT           : 443
    AWS_ACCESS_KEY_ID     : ***
    AWS_SECRET_ACCESS_KEY : ***

Shell commands:
  AWS S3 Alias          : alias s3='AWS_ACCESS_KEY_ID=***
AWS_SECRET_ACCESS_KEY=*** aws s3 --no-verify-ssl --endpoint-url ***'

Bucket status:
  Name                  : obc-test-35800e50-8978-461f-b7e0-7793080e26ba
  Type                  : REGULAR
  Mode                  : OPTIMAL
  ResiliencyStatus      : OPTIMAL
  QuotaStatus           : QUOTA_NOT_SET
  Num Objects           : 0
  Data Size             : 0.000 B
  Data Size Reduced     : 0.000 B
  Data Space Avail      : 13.261 GB
  Num Objects Avail     : 9007199254740991
```

Enabling secured mode deployment for Multicloud Object Gateway

You can specify a range of IP addresses that should be allowed to reach the Multicloud Object Gateway (MCG) load balancer services to enable secure mode deployment. This helps to control the IP addresses that can access the MCG services.

Before you begin

- A running Fusion Data Foundation cluster.
- In case of a bare metal deployment, ensure that the load balancer controller supports setting the loadBalancerSourceRanges attribute in the Kubernetes services.

Procedure

Edit the NooBaa custom resource (CR) to specify the range of IP addresses that can access the MCG services after deploying Fusion Data Foundation.

```
oc edit noobaa -n openshift-storage noobaa
```

noobaa
: The NooBaa CR type that controls the NooBaa system deployment.

noobaa
: The name of the NooBaa CR.

For example:

```
...
spec:
  ...
  loadBalancerSourceSubnets:
    s3: ["10.0.0.0/16", "192.168.10.0/32"]
    sts:
      - "10.0.0.0/16"
      - "192.168.10.0/32"
...
```

loadBalancerSourceSubnets
: A new field that can be added under **spec** in the NooBaa CR to specify the IP addresses that should have access to the NooBaa services.

  In this example, all the IP addresses that are in the subnet 10.0.0.0/16 or 192.168.10.0/32 be able to access MCG S3 and security token service (STS) while the other IP addresses are not allowed to access.

What to do next

To verify if the specified IP addresses are set, from the OpenShift Web Console, run the following command and check if the output matches with the IP addresses provided to MCG:

```
oc get svc -n openshift-storage <s3 | sts> -o=go-template='{{ .spec.loadBalancerSourceRanges }}'
```

Disabling routes that enable external access to Multicloud Object Gateway

Disable **S3**, **STS**, and **noobaa-mgmt** routes by updating the storagecluster CR.

About this task

You can disable **S3**, **STS**, and **noobaa-mgmt** routes by setting the following option in the storagecluster CR:

```
spec:
  multiCloudGateway:
    disableRoutes: true
```

This stops reconciling of all the three routes so that the routes are not recreated when they are deleted.

Procedure

1. Edit the storagecluster CR in one of the following ways:
   - Using the **oc patch** command:

     ```
     oc patch storagecluster ocs-storage -n openshift-storage --type=merge -p '{"spec":{"multiCloudGateway":
     {"disableRoutes":true}}}'
     ```

   - Using the interactive method:

     ```
     oc edit storagecluster ocs-storage -n openshift-storage
     ```

     - Add the following under **.spec**:

       ```
       multiCloudGateway:
         disableRoutes: true
       ```

2. To verify, delete the existing routes and check if they are listed:

   ```
   oc delete routes --all -n openshift-storage
   oc get routes -n openshift-storage
   ```

   The **S3**, **STS**, and **noobaa-mgmt** routes are no longer reconciled by the operator.

Mirroring data for hybrid and Multicloud buckets

Create bucket classes to mirror data for hybrid and Multicloud buckets. Mirroring data can be setup by using the OpenShift UI, YAML, or MCG command-line interface.

You can use the simplified process of the Multicloud Object Gateway (MCG) to span data across cloud providers and clusters. Before you create a bucket class that reflects the data management policy and mirroring, you must add a backing storage that can be used by the MCG. For information, see Adding storage resources for hybrid or Multicloud.

- **Creating bucket classes to mirror data using the MCG command-line-interface**
  Create bucket classes to mirror data for hybrid and Multicloud buckets using the MCG command-line interface.
- **Creating bucket classes to mirror data using a YAML**
  Create bucket classes to mirror data for hybrid and Multicloud buckets using a YAML file.

Creating bucket classes to mirror data using the MCG command-line-interface

Create bucket classes to mirror data for hybrid and Multicloud buckets using the MCG command-line interface.

Before you begin

Ensure to download Multicloud Object Gateway (MCG) command-line interface.

Procedure

1. From the Multicloud Object Gateway (MCG) command-line interface, run the following command to create a bucket class with a mirroring policy:

   ```
   odf noobaa bucketclass create placement-bucketclass mirror-to-aws --backingstores=azure-resource,aws-resource --placement
   Mirror
   ```

2. Set the newly created bucket class to a new bucket claim to generate a new bucket that is mirrored between two locations:

   ```
   odf noobaa obc create  mirrored-bucket --bucketclass=mirror-to-aws
   ```

Creating bucket classes to mirror data using a YAML

Create bucket classes to mirror data for hybrid and Multicloud buckets using a YAML file.

Procedure

1. Apply the following YAML.
   This YAML is a hybrid example that mirrors data between local Ceph storage and AWS.

```
apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: noobaa
  name: <bucket-class-name>
  namespace: openshift-storage
spec:
  placementPolicy:
    tiers:
    - backingStores:
      - <backing-store-1>
      - <backing-store-2>
      placement: Mirror
```

2. Add the following lines to your standard Object Bucket Claim (OBC):

```
additionalConfig:
  bucketclass: mirror-to-aws
```

   For more information about OBCs, see Object Bucket Claim.

Bucket policies in the Multicloud Object Gateway

Fusion Data Foundation supports AWS S3 bucket policies. Bucket policies allow you to grant users access permissions for buckets and the objects in them. Use the information in this section to understand how to use bucket policies in Multicloud Object Gateway.

- **Introduction to bucket policies**
  Bucket policies are an access policy option available for you to grant permission to your AWS S3 buckets and objects. Bucket policies use JSON-based access policy language.
- **Using bucket policies in Multicloud Object Gateway**
  Use these instructions to use bucket policies in Multicloud Object Gateway.
- **Creating a user in the Multicloud Object Gateway**
  Create a user account in the Multicloud Object Gateway using the MCG command-line interface, optionally with a NamespaceStore file system configuration.

Introduction to bucket policies

Bucket policies are an access policy option available for you to grant permission to your AWS S3 buckets and objects. Bucket policies use JSON-based access policy language.

For more information about access policy language, see AWS Access Policy Language Overview.

Using bucket policies in Multicloud Object Gateway

Use these instructions to use bucket policies in Multicloud Object Gateway.

Before you begin

Ensure you have the following:

- A running Fusion Data Foundation Platform.
- Access to the Multicloud Object Gateway (MCG), see Accessing the Multicloud Object Gateway with your applications.

Procedure

1. Create the bucket policy in JSON format.
   For example:

```
{
    "Version": "NewVersion",
    "Statement": [
        {
            "Sid": "Example",
            "Effect": "Allow",
            "Principal": [
                    "john.doe@example.com"
            ],
            "Action": [
                "s3:GetObject"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::john_bucket"
            ]
        }
    ]
}
```

2. Using AWS S3 client, use the **put-bucket-policy** command to apply the bucket policy to your S3 bucket:

   ```
   aws --endpoint ENDPOINT --no-verify-ssl s3api put-bucket-policy --bucket MyBucket --policy BucketPolicy
   ```

   ENDPOINT
   : An S3 endpoint.

   MyBucket
   : A name of the bucket to set the policy on.

   BucketPolicy
   : A bucket policy JSON file.

   --no-verify-ssl
   : Add this if you are using the default self signed certificates.

   For example:

   ```
   aws --endpoint https://s3-openshift-storage.apps.gogo44.noobaa.org --no-verify-ssl s3api put-bucket-policy -bucket MyBucket --policy file://BucketPolicy
   ```

   For more information on the **put-bucket-policy** command, see AWS CLI Command Reference for put-bucket-policy.

   Note: The principal element specifies the user that is allowed or denied access to a resource, such as a bucket. Currently, only NooBaa accounts can be used as principals. In the case of object bucket claims, NooBaa automatically creates an account .

   ```
   obc-account.<generated bucket name>@noobaa.io
   ```

   Note: Bucket policy conditions are not supported.

What to do next

There are many available elements for bucket policies with regard to access permissions.

- For details on these elements and examples of how they can be used to control the access permissions, see AWS Access Policy Language Overview.
- For more examples of bucket policies, see AWS Bucket Policy Examples.
- Fusion Data Foundation version 4.16 or higher introduces the bucket policy elements `NotPrincipal`, `NotAction`, and `NotResource`. For more information on these elements, see IAM JSON policy elements reference.

Creating a user in the Multicloud Object Gateway

Create a user account in the Multicloud Object Gateway using the MCG command-line interface, optionally with a NamespaceStore file system configuration.

Before you begin

- A running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager.
  - For IBM Power, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
    ```

  - For IBM Z infrastructure, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
    ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

Procedure

Run the following command to create an MCG user account:

```
odf noobaa account create <noobaa-account-name> [--allow_bucket_create=true] [--default_resource=''] [--force_md5_etag=false] [--gid=-1] [--new_buckets_path='/'] [--nsfs_account_config=false] [--nsfs_only=true] [--uid=-1]
```

<allow_bucket_create>
: Specify if this account be allowed to create new buckets.

<default_resource>
: Specify the default resource on which new buckets are created.

<force_md5_etag>
Specify if **`md5 etag`** calculation be enabled for the account.

<gid_number>
Specify the GID number.

<new_buckets_path>
Specify the path where the new buckets are created.

<nsfs_account_config>
Specify if NSFS account needs to be created.

<nsfs_only>
Set if this account is used only for NSFS.

<uid_number>
Specify the UID number.

Bucket notification in Multicloud Object Gateway

With bucket notification, you can send a notification to an external server whenever there is an event in the bucket.

Multicloud Object Gateway (MCG) supports several bucket event notification types. You can receive bucket notifications for activities such as flow creation, triggering data digestion, and so on. MCG supports the following event types that can be specified in the notification configuration:

- s3:TestEvent
- s3:ObjectCreated:*
- s3:ObjectCreated:Put
- s3:ObjectCreated:Post
- s3:ObjectCreated:Copy
- s3:ObjectCreated:CompleteMultipartUpload
- s3:ObjectRemoved:*
- s3:ObjectRemoved:Delete
- s3:ObjectRemoved:DeleteMarkerCreated
- s3:LifecycleExpiration:*
- s3:LifecycleExpiration:Delete
- s3:LifecycleExpiration:DeleteMarkerCreated
- s3:ObjectRestore:*
- s3:ObjectRestore:Post
- s3:ObjectRestore:Completed
- s3:ObjectRestore:Delete
- s3:ObjectTagging:*
- s3:ObjectTagging:Put
- s3:ObjectTagging:Delete

- **Configuring bucket notification in Multicloud Object Gateway**
  Learn how to configure bucket notifications in Multicloud Object Gateway (MCG).

Configuring bucket notification in Multicloud Object Gateway

Learn how to configure bucket notifications in Multicloud Object Gateway (MCG).

Before you begin

- Ensure to have one of the following before configuring:
  - Kafka cluster is deployed.

    For example, you can deploy Kafka by using **`AMQ/strimzi`** by referring to Getting Started with AMQ Streams on OpenShift.

  - An HTTP or HTTPs server is connected.

    For example, you can set up an HTTP server to log incoming HTTP requests so that you can observe them using the **`oc logs`** command as follows:

```
cat http_logging_server.yaml
apiVersion: v1
kind: List
metadata: {}
items:
- apiVersion: apps/v1
  kind: Deployment
  metadata:
    name: http-logger
  spec:
    replicas: 1
    selector:
      matchLabels:
        app: http-logger
    template:
      metadata:
```

```
        labels:
          app: http-logger
      spec:
        containers:
        - name: http-logger
          image: registry.redhat.io/ubi9/python-39:latest
          command:
            - /bin/sh
            - -c
            - |
              set -e  # Fail on any error
              mkdir -p /tmp/app
              pip install flask
              cat <<EOF > /tmp/app/server.py
              from flask import Flask, request
              app = Flask(__name__)
              @app.route("/", methods=["POST", "GET"])
              def log_request():
                  body = request.get_data(as_text=True)
                  print(body)  # Simple one-line logging per request
                  return "", 200
              if __name__ == "__main__":
                  app.run(host="0.0.0.0", port=8676, debug=True)
              EOF
              exec python /tmp/app/server.py
          ports:
          - containerPort: 8676
            protocol: TCP
          securityContext:
            runAsNonRoot: true
            allowPrivilegeEscalation: false
- apiVersion: v1
  kind: Service
  metadata:
    name: http-logger
    labels:
      app: http-logger
  spec:
    selector:
      app: http-logger
    ports:
      - port: 8676
        targetPort: 8676
        protocol: TCP
        name: http
```

```
oc create -f http_logging_server.yaml -n <http-server-namespace>
```

- Ensure that server is accessible from the MCG core pod from which the notifications are sent.
- Ensure to fetch MCG's credentials:

```
NOOBAA_ACCESS_KEY=$(oc extract secret/noobaa-admin -n openshift-storage --keys=AWS_ACCESS_KEY_ID --to=- 2>/dev/null); \
NOOBAA_SECRET_KEY=$(oc extract secret/noobaa-admin -n openshift-storage --keys=AWS_SECRET_ACCESS_KEY --to=- 2>/dev/null);
\
S3_ENDPOINT=https://$(oc get route s3 -n openshift-storage -o json | jq -r ".spec.host")
alias aws_alias='AWS_ACCESS_KEY_ID=$NOOBAA_ACCESS_KEY AWS_SECRET_ACCESS_KEY=$NOOBAA_SECRET_KEY aws --endpoint $S3_ENDPOINT
--no-verify-
```

Procedure

1. Describe connection using a json file, for example, connection.json on your local machine.
   - **An example for Kafka**:

     ```
     {
       "name": "kafka_notif_conn_file" <-- any string
       "notification_protocol": "kafka",
       "metadata.broker.list": "<kafka-service-name>.<project>.svc.cluster.local:<kafka-service-port>",
       "topic": "my-topic",<--  refer to an existing KafkaTopic resource
     }
     ```

     The structure under `metadata.broker.list` must be `<service-name>.<namespace>.svc.cluster.local:9092`, topic must refer to the name of an existing `kafkatopic` resource in the namespace.

   - **An example for HTTP or HTTPs**:

     ```
     {
       "name": "http_notif_connection_config",
       "notification_protocol": "http", <-- or "https"
       "agent_request_object": {
         "host": "<http-service>.<http-server-namespace>.svc.cluster.local",
         "port": <http-server-port>
        }
     }
     ```

   Additional options:

   request_options_object
   : Value is a JSON that is passed to nodejs' http(s) request (optional).
   Any field supported by nodejs' http(s) request option can be used, for example:

   path'
   : Used to specify the url path.

   'auth'

Used for http simple auth. Syntax for the value of 'auth' is: <name>:<password>.

2. Create a secret from the file in the **openshift-storage** namespace.
   For example:

```
oc create secret generic <connection-secret> --from-file=connect.json -n openshift-storage
```

3. Update the **NooBaa** CR in the **openshift-storage** namespace with the connection secret and an optional CephFS RWX PVC.
   For example:

```
oc patch noobaa noobaa --type='merge' -n openshift-storage -p '{
  "spec": {
    "bucketNotifications": {
      "connections": [
        {
          "name": <connection-secret>,
          "namespace": "openshift-storage"
        }
      ],
      "enabled": true,
      "pvc": "bn-pvc" <-- optional
    }
  }
}'
```

   Note: If PVC is not provided, MCG creates one automatically as needed.
   For example:

```
oc get pvc bn-pvc -n openshift-storage
NAME     STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS                 VOLUMEATTRIBUTESCLASS   AGE
bn-pvc   Bound    pvc-24683f8d-48e4-4c6b-b108-d507ca2f4fd1   25Gi       RWX            ocs-storagecluster-cephfs   <unset>                 25h
```

   Important: Names of the connection secret must be unique in the list even though the namespaces are different.
   Wait for the **noobaa-core** and **noobaa-endpoint** pods to restart before proceeding with the next step.

4. Use **S3:PutBucketNotification** on an MCG bucket by using the **noobaa-admin** credentials and S3 endpoint under an S3 alias.
   For example:

```
aws_alias s3api put-bucket-notification --bucket first.bucket --notification-configuration '{
  "TopicConfiguration": {
    "Id": "<notif_event_kafka>", <-- Unique string
    "Events": ["s3:ObjectCreated:*"], <-- a filter for events
    "Topic": "<connection-secret/connect.json>"
  }
}'
```

5. Verify that the bucket notification configuration is set on the bucket.
   For example:

```
aws_alias s3api get-bucket-notification-configuration --bucket first.bucket
{
    "TopicConfigurations": [
        {
            "Id": "notif_event_kafka",
            "TopicArn": "kafka-connection-secret/connect.json",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

6. Add some objects on the bucket.

```
echo 'a' | aws_alias s3 cp - s3://first.bucket/a
```

   Wait for a while and query the topic messages to verify the expected notification has been sent and received.

   An example for Kafka:

```
oc -n your-kafka-project rsh my-cluster-kafka-0 bin/kafka-console-consumer.sh \
  --bootstrap-server my-cluster-kafka-bootstrap.myproject.svc.cluster.local:9092 \
  --topic my-topic \
  --from-beginning --timeout-ms 10000 | grep '^{.*}' | jq -c '.' | jq
```

   An example for HTTP or HTTPs:

```
oc logs deployment/http-logger -n <http-server-namespace> | grep '^{.*}' | jq
```

   Example output:

```
{
  "Records": [
    {
      "eventVersion": "2.3",
      "eventSource": "noobaa:s3",
      "eventTime": "2024-11-27T12:44:21.987Z",
      "s3": {
        "s3SchemaVersion": "1.0",
        "object": {
          "sequencer": 10,
          "key": "a",
```

```
            "eTag": "60b725f10c9c85c70d97880dfe8191b3"
          },
          "bucket": {
            "name": "second.bucket",
            "ownerIdentity": {
              "principalId": "admin@noobaa.io"
            },
            "arn": "arn:aws:s3:::first.bucket"
          }
        },
        "eventName": "ObjectCreated:Put",
        "userIdentity": {
          "principalId": "noobaa"
        },
        "requestParameters": {
          "sourceIPAddress": "100.64.0.3"
        },
        "responseElements": {
          "x-amz-request-id": "m3zvo0cm-5239xd-bhj",
          "x-amz-id-2": "m3zvo0cm-5239xd-bhj"
        }
      }
    ]
}
```


Multicloud Object Gateway bucket replication

Data replication from one Multicloud Object Gateway (MCG) bucket to another MCG bucket provides higher resiliency and better collaboration options. These buckets can be either data buckets or namespace buckets backed by any supported storage solution (S3, Azure, and so on). Replicate a bucket to another bucket and set replication policies using command-line interface and YAML.

A replication policy is composed of a list of replication rules. Each rule defines the destination bucket, and can specify a filter based on an object key prefix. Configuring a complementing replication policy on the second bucket results in bidirectional replication.

- **Replicating a bucket to another bucket**
  This section describes how to replicate one bucket to another using CLI and YAML.
- **Setting a bucket class replication policy**
  Set a bucket class replication policy to automatically replicate all buckets created under a bucket class using the MCG command-line interface or a YAML file.
- **Enabling log based bucket replication**
  This section provides information about the log based bucket replication.
- **Bucket logging for Multicloud Object Gateway**
  Bucket logging helps in recording the S3 operations that are performed against the Multicloud Object Gateway (MCG) bucket for compliance, auditing, and optimization purposes.
- **Synchronizing versions in Multicloud Object Gateway bucket replication**
  Learn how to synchronize versions in Multicloud Object Gateway (MCG) bucket replication.
- **Obtaining metrics to reflect bucket replication state**
  Learn how to obtain bucket-level replication metrics to evaluate replication progress.

Replicating a bucket to another bucket

This section describes how to replicate one bucket to another using CLI and YAML.

You can set the bucket replication policy in two ways:

- Replicating a bucket to another bucket using the MCG command-line interface.
- Replicating a bucket to another bucket using a YAML.

- **Replicating a bucket to another bucket using the MCG command-line interface**
  Provide higher resiliency and better collaboration options by replicating a bucket to another bucket using the MCG command-line interface.
- **Replicating a bucket to another bucket using a YAML**
  Provide higher resiliency and better collaboration options by replicating a bucket to another bucket using the a YAML file.

Replicating a bucket to another bucket using the MCG command-line interface

Provide higher resiliency and better collaboration options by replicating a bucket to another bucket using the MCG command-line interface.

Before you begin

- Ensure you have a running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager. In case of IBM Z infrastructure, use the following command:

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

About this task

Applications that require a Multicloud Object Gateway (MCG) bucket to have a specific replication policy can create an Object Bucket Claim (OBC) and define the replication policy parameter in a JSON file.

Procedure

From the MCG command-line interface, run the following command to create an OBC with a specific replication policy:

```
odf noobaa obc create <bucket-claim-name> -n openshift-storage --replication-policy /path/to/json-file.json
```

bucket-claim-name
Specify the name of the bucket claim.

/path/to/json-file.json
Is the path to a JSON file which defines the replication policy.

Example JSON file:

```
[{ "rule_id": "rule-1", "destination_bucket": "first.bucket", "filter": {"prefix": "repl"}}]
```

`"prefix"`
(Optional:) It is the prefix of the object keys that should be replicated, and you can even leave it empty, for example, `{"prefix": ""}`.
For example:

```
odf noobaa obc create my-bucket-claim -n openshift-storage --replication-policy /path/to/json-file.json
```

Replicating a bucket to another bucket using a YAML

Provide higher resiliency and better collaboration options by replicating a bucket to another bucket using the a YAML file.

Before you begin

- Ensure you have a running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager. In case of IBM Z infrastructure, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
  ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

About this task

Applications that require a Multicloud Object Gateway (MCG) data bucket to have a specific replication policy can create an Object Bucket Claim (OBC) and add the spec.additionalConfig.replication-policy parameter to the OBC. For more information about OBCs, see Object Bucket Claim.

Procedure

Apply the following YAML:

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: <desired-bucket-claim>
  namespace: <desired-namespace>
spec:
  generateBucketName: <desired-bucket-name>
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    replication-policy: [{ "rule_id": "<rule id>", "destination_bucket": "first.bucket", "filter": {"prefix": "<object name prefix>"}}]
```

desired-bucket-claim
Specify the name of the bucket claim.

desired-namespace
Specify the namespace.

desired-bucket-name>
Specify the prefix of the bucket name.

rule_id
Specify the ID number of the rule, for example, `{"rule_id": "rule-1"}`.

destination_bucket
Specify the name of the destination bucket, for example, `{"destination_bucket": "first.bucket"}`.

object name prefix
(Optional:) It is the prefix of the object keys that should be replicated, and you can even leave it empty, for example, `{"prefix": ""}`.

Setting a bucket class replication policy

Set a bucket class replication policy to automatically replicate all buckets created under a bucket class using the MCG command-line interface or a YAML file.

It is possible to set up a replication policy that automatically applies to all the buckets created under a certain bucket class. You can do this in two ways:

- Setting a bucket class replication policy using the MCG command-line interface.
- Setting a bucket class replication policy using a YAML.

- **Setting a bucket class replication policy using the MCG command-line interface**
  Provide higher resiliency and better collaboration options by setting a bucket class replication policy using the MCG command-line interface.
- **Setting a bucket class replication policy using a YAML**
  Provide higher resiliency and better collaboration options by setting a bucket class replication policy using a YAML file.

Setting a bucket class replication policy using the MCG command-line interface

Provide higher resiliency and better collaboration options by setting a bucket class replication policy using the MCG command-line interface.

Before you begin

- Ensure you have a running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager. In case of IBM Z infrastructure, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
  ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

About this task

Applications that require a Multicloud Object Gateway (MCG) bucket class to have a specific replication policy can create a `bucketclass` and define the replication-policy parameter in a JSON file.

It is possible to set a bucket class replication policy for two types of bucket classes:

- Placement
- Namespace

Procedure

From the MCG command-line interface, run the following command:

```
odf noobaa -n openshift-storage bucketclass create placement-bucketclass <bucketclass-name> --backingstores <backingstores> --replication-policy=/path/to/json-file.json
```

bucketclass-name
Specify the name of the bucket class.

backingstores
Specify the name of a backingstore. It is possible to pass several backingstores separated by commas.

/path/to/json-file.json
Is the path to a JSON file which defines the replication policy.

Example JSON file:

```
[{ "rule_id": "rule-1", "destination_bucket": "first.bucket", "filter": {"prefix": "repl"}}]
```

"prefix"
(Optional:) It is the prefix of the object keys that should be replicated, and you can even leave it empty, for example, `{"prefix": ""}`.
For example:

```
noobaa -n openshift-storage bucketclass create placement-bucketclass bc --backingstores azure-blob-ns --replication-polic
```

This example creates a placement bucket class with a specific replication policy defined in the JSON file.

Setting a bucket class replication policy using a YAML

Provide higher resiliency and better collaboration options by setting a bucket class replication policy using a YAML file.

Before you begin

- Ensure you have a running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager. In case of IBM Z infrastructure, use the following command:

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
  ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

About this task

Applications that require a Multicloud Object Gateway (MCG) bucket class to have a specific replication policy can create a bucket class using the spec.replicationPolicy field.

Procedure

Apply the following YAML:

```
apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: <desired-app-label>
  name: <desired-bucketclass-name>
  namespace: <desired-namespace>
spec:
  placementPolicy:
    tiers:
    - backingstores:
      - <backingstore>
      placement: Spread
  replicationPolicy: [{ "rule_id": "<rule id>", "destination_bucket": "first.bucket", "filter": {"prefix": "<object name
prefix>"}}]
```

This YAML is an example that creates a placement bucket class. Each Object bucket claim (OBC) object that is uploaded to the bucket is filtered based on the prefix and is replicated to `first.bucket`.

desired-app-label
Specify a label for the app.

desired-bucketclass-name
Specify the bucket class name.

desired-namespace
Specify the namespace in which the bucket class gets created.

backingstore
Specify the name of a backingstore. It is possible to pass several backingstores.

rule_id
Specify the ID number of the rule, for example, `{"rule_id": "rule-1"}`.

destination_bucket
Specify the name of the destination bucket, for example, `{"destination_bucket": "first.bucket"}`.

object name prefix
(Optional:) It is the prefix of the object keys that should be replicated, and you can even leave it empty, for example, `{"prefix": ""}`.

Enabling log based bucket replication

This section provides information about the log based bucket replication.

When creating a bucket replication policy, you can use logs so that recent data is replicated more quickly, while the default scan-based replication works on replicating the rest of the data.

Important:
This feature requires setting up bucket logs on AWS or Azure. For more information about setting up AWS logs, see Enabling Amazon S3 server access logging. The AWS logs bucket needs to be created in the same region as the source NamespaceStore AWS bucket.

Note:
This feature is only supported in buckets that are backed by a NamespaceStore. Buckets backed by BackingStores cannot utilize log-based replication.

- **Enabling log based bucket replication for new namespace buckets using OpenShift Web Console in AWS environment**
  You can optimize replication by using the event logs of the Amazon Web Service(AWS) cloud environment. You enable log based bucket replication for new namespace buckets using the web console during the creation of namespace buckets.
- **Enabling log based bucket replication for existing namespace buckets using YAML**
  Enable log-based bucket replication for existing namespace buckets by editing the OBC YAML to add a replication policy with a destination bucket and a log bucket location.
- **Enabling log based bucket replication in Microsoft Azure**
  Enable log-based bucket replication to a Microsoft Azure namespace store by registering an Azure application, creating the required NamespaceStore, and applying the replication policy.
- **Enabling log-based bucket replication deletion**
  Enable deletion synchronization for log-based bucket replication so that objects deleted in the source bucket are also removed from the destination bucket.

Enabling log based bucket replication for new namespace buckets using OpenShift Web Console in AWS environment

You can optimize replication by using the event logs of the Amazon Web Service(AWS) cloud environment. You enable log based bucket replication for new namespace buckets using the web console during the creation of namespace buckets.

Before you begin

- Ensure that object logging is enabled in AWS. For more information, see the “Using the S3 console” section in Enabling Amazon S3 server access logging.
- Administrator access to OpenShift Web Console.

About this task

Procedure

1. In the OpenShift Web Console, navigate to **Storage** → **Object Storage** → **Object Bucket Claims**.
2. Click **Create ObjectBucketClaim**.
3. Enter the name of ObjectBucketName and select StorageClass and BucketClass.
4. Select the **Enable replication** check box to enable replication.
5. In the **Replication policy** section, select the **Optimize replication using event logs** check box.
6. Enter the name of the bucket that contains the logs under **Event log Bucket**.
   If the logs are not stored in the root of the bucket, provide the full path without `s3://`
7. Enter a prefix to replicate only the objects whose name begins with the given prefix.

Enabling log based bucket replication for existing namespace buckets using YAML

Enable log-based bucket replication for existing namespace buckets by editing the OBC YAML to add a replication policy with a destination bucket and a log bucket location.

About this task

You can enable log based bucket replication for the existing buckets that are created using the command line interface or by applying an YAML, and not the buckets that are created using AWS S3 commands.

Procedure

Edit the YAML of the bucket's OBC to enable log based bucket replication. Add the following under `spec`:

```
replicationPolicy: '{"rules":[{"rule_id":"<RULE ID>", "destination_bucket":"<DEST>", "filter": {"prefix": "<PREFIX>"}}],
"log_replication_info": {"logs_location": {"logs_bucket": "<LOGS_BUCKET>"}}}'
```

`rule_id`
: Specify an ID of your choice for identifying the rule

`destination_bucket`
: Specify the name of the target MCG bucket that the objects are copied to

(optional) `{"filter": {"prefix": <>}}`
: Specify a prefix string that you can set to filter the objects that are replicated

`log_replication_info`
Specify an object that contains data related to log-based replication optimization. `{"logs_location": {"logs_bucket": <>}}` is set to the location of the AWS S3 server access logs.

Enabling log based bucket replication in Microsoft Azure

Enable log-based bucket replication to a Microsoft Azure namespace store by registering an Azure application, creating the required NamespaceStore, and applying the replication policy.

Before you begin

- Refer to Microsoft Azure documentation and ensure that you have completed the following tasks in the Microsoft Azure portal:
  1. Ensure that have created a new application and noted down the name, application (client) ID, and directory (tenant) ID.

     For information, see Register an application.
  2. Ensure that a new client secret is created and the application secret is noted down.
  3. Ensure that a new Log Analytics workspace is created and its name and workspace ID is noted down.

     For information, see Create a Log Analytics workspace.
  4. Ensure that the `Reader` role is assigned under **Access control** and members are selected and the name of the application that you registered in the previous step is provided.

     For more information, see Assign Azure roles using the Azure portal.
  5. Ensure that a new storage account is created and the Access keys are noted down.
  6. In the **Monitoring** section of the storage account created, select a blob and in the **Diagnostic settings** screen, select only `StorageWrite` and `StorageDelete`, and in the destination details add the Log Analytics workspace that you created earlier. Ensure that a blob is selected in the **Diagnostic settings** screen of the **Monitoring** section of the storage account created. Also, ensure that only `StorageWrite` and `StorageDelete` is selected and in the destination details, the Log Analytics workspace that you created earlier is added.

     For more information, see Diagnostic settings in Azure Monitor.
  7. Ensure that two new containers for object source and object destination are created.
- Administrator access to OpenShift Web Console.

Procedure

1. Create a secret with credentials to be used by the `namespacestores`.

```
apiVersion: v1
kind: Secret
metadata:
  name: <namespacestore-secret-name>
type: Opaque
data:
    TenantID: <AZURE TENANT ID ENCODED IN BASE64>
    ApplicationID: <AZURE APPLICATIOM ID ENCODED IN BASE64>
    ApplicationSecret: <AZURE APPLICATION SECRET ENCODED IN BASE64>
    LogsAnalyticsWorkspaceID: <AZURE LOG ANALYTICS WORKSPACE ID ENCODED IN BASE64>
    AccountName: <AZURE ACCOUNT NAME ENCODED IN BASE64>
    AccountKey: <AZURE ACCOUNT KEY ENCODED IN BASE64>
```

2. Create a `NamespaceStore` backed by a container created in Azure.
   For more information, see Adding a namespace bucket using the OpenShift Container Platform user interface.
3. Create a new Namespace-Bucketclass and OBC that utilizes it.
4. Check the object bucket name by looking in the YAML of target OBC, or by listing all S3 buckets, for example, `- s3 ls`.
5. Use the following template to apply an Azure replication policy on your source OBC by adding the following in its YAML, under `.spec`:

```
replicationPolicy:'{"rules":[ {"rule_id":"ID goes here", "sync_deletions": "<true or false>"", "destination_bucket":object
bucket name"}
 ], "log_replication_info":{"endpoint_type":"AZURE"}}'
```

`sync_deletion`
Specify a boolean value, `true` or `false`.
`destination_bucket`
Ensure to use the name of the object bucket, and not the claim. The name can be retrieved using the `s3 ls` command, or by looking for the value in an OBC's YAML.

What to do next

- **Verification steps:**
  1. Write objects to the source bucket.
  2. Wait until MCG replicates them.
  3. Delete the objects from the source bucket.
  4. Verify the objects were removed from the target bucket.

Enabling log-based bucket replication deletion

Enable deletion synchronization for log-based bucket replication so that objects deleted in the source bucket are also removed from the destination bucket.

Before you begin

- Administrator access to OpenShift Web Console.
- AWS Server Access Logging configured for the desired bucket.

Procedure

1. In the OpenShift Web Console, navigate to Storage > Object Storage > Object Bucket Claims.
2. Click Create new Object bucket claim.
3. Optional: In the Replication rules section, select the Sync deletion checkbox for each rule separately.
4. Enter the name of the bucket that contains the logs under Event log Bucket.
   If the logs are not stored in the root of the bucket, provide the full path without `s3://`.
5. Enter a prefix to replicate only the objects whose name begins with the given prefix.

Bucket logging for Multicloud Object Gateway

Bucket logging helps in recording the S3 operations that are performed against the Multicloud Object Gateway (MCG) bucket for compliance, auditing, and optimization purposes.

Bucket logging supports the following two options:

- Best-Effort: Bucket logging is recorded by using the Unified Data Platform (UDP) on a best-effort basis.
- Guaranteed: Bucket logging with this option creates a PersistentVolumeClaim (PVC) attached to the MCG pods and saves the logs to this PVC on a guaranteed basis, and then from the PVC to the log buckets. When this option is used, logging occurs twice for each S3 operation, as follows:
  - At the start of processing the request.
  - At the end, with the result of the S3 operation.

- **Enabling bucket logging by using the Best-Effort option**
  Use the NooBaa API or the S3 API to enable bucket logging for the best-effort option.
- **Enabling bucket logging by using the Guaranteed option**
  Use the default CephFS storage class or ReadWriteMany PersistentVolumeClaim (RWX PVC) to enable bucket logging for the guaranteed option.

Enabling bucket logging by using the Best-Effort option

Use the NooBaa API or the S3 API to enable bucket logging for the best-effort option.

Before you begin

- Ensure that you installed the Openshift Container Platform with the Fusion Data Foundation operator.
- Ensure that you have access to MCG.

  For more information, see Accessing the Multicloud Object Gateway with your applications.

Procedure

1. Create a data bucket where you can upload the objects.

   ```
   odf nb bucket create data.bucket
   ```

2. Create a log bucket where you want to store the logs for bucket operations.

   ```
   odf nb bucket create log.bucket
   ```

3. Configure bucket logging on data bucket with log bucket.
   You can configure bucket logging by using the NooBaa API or the S3 API.

   - By using the NooBaa API:

     ```
     odf nb api bucket_api put_bucket_logging '{
        "name": "data.bucket",
        "log_bucket": "log.bucket",
        "log_prefix": "data-bucket-logs"
     }'
     ```

   - By using the S3 API:

     - Define an alias for the S3 API command.

       ```
       alias s3api_alias='AWS_ACCESS_KEY_ID=$NOOBAA_ACCESS_KEY AWS_SECRET_ACCESS_KEY=$NOOBAA_SECRET_KEY aws --endpoint
       https://localhost:10443 --no-verify-ssl s3api'
       ```

- Create a setlogging.json file in the following format:

```
{
  "LoggingEnabled": {
     "TargetBucket": "<log-bucket-name>",
     "TargetPrefix": "<prefix/empty-string>"
  }
}
```

- Configure bucket logging based on settings defined in the setlogging.json file.

```
s3api_alias put-bucket-logging --endpoint <ep> --bucket <source-bucket> --bucket-logging-status
file://setlogging.json --no-verify-ssl
```

4. Verify whether the bucket logging is set for the data bucket in one of the following ways:
   - By using the NooBaa API:

```
odf nb api bucket_api get_bucket_logging '{
   "name": "data.bucket"
}'
```

   - By using the S3 API:

```
s3api_alias get-bucket-logging --no-verify-ssl --endpoint <ep> --bucket <source-bucket>
```

     The S3 operations can take up to 24 hours to get recorded in the logs bucket. The following example shows the recorded logs and how to download them:

```
s3_alias cp s3://logs.bucket/data-bucket-logs/logs.bucket.bucket_data-bucket-logs_1719230150.log - | tail -n 2
```

```
Jun 24 14:00:02 10-XXX-X-XXX.sts.openshift-storage.svc.cluster.local
{"noobaa_bucket_logging":"true","op":"GET","bucket_owner":"operator@noobaa.io","source_bucket":"data.bucket","object_
key":"/data.bucket?list-type=2&prefix=data-bucket-logs&delimiter=%2F&encoding-
type=url","log_bucket":"logs.bucket","remote_ip":"100.XX.X.X","request_uri":"/data.bucket?list-type=2&prefix=data-
bucket-logs&delimiter=%2F&encoding-type=url","request_id":"luv2XXXX-ctyg2k-12gs"} Jun 24 14:00:06 10-XXX-X-
XXX.s3.openshift-storage.svc.cluster.local
{"noobaa_bucket_logging":"true","op":"PUT","bucket_owner":"operator@noobaa.io","source_bucket":"data.bucket","object_
key":"/data.bucket/B69EC83F-0177-44D8-A8D1-
4A10C5A5AB0F.file","log_bucket":"logs.bucket","remote_ip":"100.XX.X.X","request_uri":"/data.bucket/B69EC83F-0177-
44D8-A8D1-4A10C5A5AB0F.file","request_id":"luv2XXXX-9syea5-x5z"}
```

5. Optional: To disable bucket logging, run the following command:

```
odf nb api bucket_api delete_bucket_logging '{
   "name": "data.bucket"
}'
```

Enabling bucket logging by using the Guaranteed option

Use the default CephFS storage class or ReadWriteMany PersistentVolumeClaim (RWX PVC) to enable bucket logging for the guaranteed option.

Enable guaranteed bucket logging by using the NooBaa Custom Resource (CR) in one of the following ways:

- By using the default Cepheus storage class:

```
bucketLogging:
{
loggingType: guaranteed
}
```

- By using the RWX PVC that you created:

  Important: Ensure that the PVC supports RWX.

```
bucketLogging:
{
loggingType: guaranteed
bucketLoggingPVC: <pvc-name>
}
```

Synchronizing versions in Multicloud Object Gateway bucket replication

Learn how to synchronize versions in Multicloud Object Gateway (MCG) bucket replication.

Before you begin

- A Multicloud Object Gateway (MCG) source bucket, which is created from an object bucket claim (OBC) and any MCG target bucket. For example, you can create the two buckets using OBCs using the MCG command line interface (CLI):

  Create a source bucket using an OBC:

```
odf noobaa obc create source-bucket --exact
odf noobaa obc create target-bucket --exact
```

  Where, `--exact` is optional.

- Ensure that S3 client aliases with MCG credentials and endpoint are set up.

```
NOOBAA_ACCESS_KEY=$(oc extract secret/noobaa-admin -n openshift-storage --keys=AWS_ACCESS_KEY_ID --to=- 2>/dev/null); \
NOOBAA_SECRET_KEY=$(oc extract secret/noobaa-admin -n openshift-storage --keys=AWS_SECRET_ACCESS_KEY --to=- 2>/dev/null);
\
S3_ENDPOINT=https://$(oc get route s3 -n openshift-storage -o json | jq -r ".spec.host")

alias common_s3='AWS_ACCESS_KEY_ID=$NOOBAA_ACCESS_KEY AWS_SECRET_ACCESS_KEY=$NOOBAA_SECRET_KEY aws --endpoint $S3_ENDPOINT
--no-verify-ssl'; \
alias s3_alias='common_s3 s3'; \
alias s3api_alias='common_s3 s3api'
```

- Ensure to enable versioning on both the source and target bucket by using the `put-bucket-versioning` command in the AWS S3 client:

```
s3api_alias put-bucket-versioning --bucket source-bucket --versioning-configuration Status=Enabled

s3api_alias put-bucket-versioning --bucket target-bucket --versioning-configuration Status=Enabled
```

Procedure

1. Patch or edit a replication policy with `sync_versions: true` to the source bucket's OBC:

   ```
   oc patch obc source-bucket -n openshift-storage --type=merge -p '{"spec": {"additionalConfig": {"replicationPolicy": "
   {\"rules\":[{\"rule_id\":\"replication-rule-a\",\"destination_bucket\":\"target-bucket\", \"sync_versions\": true}]}"}}}'
   ```

   Normal bucket replication replicates only the latest version, and the use of `sync_versions` add the functionality of replicating even the older versions, and in their original order.

   Note:
   - Delete markers are replicated if configured and using log base replication.
   - Delete version IDs are not replicated to avoid human errors.
2. Verify the replication to the target bucket by adding a few versions under an object key and waiting for them to get replicated to the target bucket:

   ```
   echo 'version_a' | s3_alias cp - s3://source-bucket/versioned_obj.txt

   echo 'version_b' | s3_alias cp - s3://source-bucket/versioned_obj.txt

   echo 'version_c' | s3_alias cp - s3://source-bucket/versioned_obj.txt
   ```

   After a few minutes, compare the versions of the object on both the buckets:

   ```
   s3api_alias list-object-versions --bucket source-bucket --prefix versioned_obj.txt | jq -r ".Versions[].ETag"
   "aaabf266d38a8e995cef03c13ee9a7f1"
   "181d8a23f59939c0cddec1692e05cdf3"
   "ebc538cc6ffa04a39263f4b1be2f832f"

   s3api_alias list-object-versions --bucket target-bucket --prefix versioned_obj.txt | jq -r ".Versions[].ETag"
   "aaabf266d38a8e995cef03c13ee9a7f1"
   "181d8a23f59939c0cddec1692e05cdf3"
   "ebc538cc6ffa04a39263f4b1be2f832f"
   ```

Obtaining metrics to reflect bucket replication state

Learn how to obtain bucket-level replication metrics to evaluate replication progress.

About this task

To determine if the data is safe or available on the secondary site, the replication progress per bucket is provided using the following metrics:

- Total number of objects scanned in the last replication cycle per bucket - `bucket_last_cycle_total_objects_num{bucket="bucket-name"}`
- Number of objects successfully replicated in the last cycle per bucket - `bucket_last_cycle_replicated_objects_num{bucket="bucket-name"}`
- Number of objects that failed replication in the last cycle per bucket - `bucket_last_cycle_error_objects_num{bucket="bucket-name"}`

Procedure

1. Use the following commands to obtain the metrics:

   ```
   JWT_TOKEN=$(oc get secret noobaa-metrics-auth-secret -n openshift-storage -o jsonpath=" {.data.metrics_token}
   " | base64 -d)

   oc -n openshift-storage exec noobaa-core-0 – curl -k -H "Authorization: Bearer ${JWT_TOKEN}"
   localhost:8080/metrics/bg_workers
   ```

   For example:

   ```
   noobaa-operator$ oc -n openshift-storage exec -it noobaa-core-0 -- curl -k -H "Authorization: Bearer ${JWT_TOKEN}"
   http://localhost:8080/metrics/bg_workers | grep -E "bucket_name"| grep -v "NooBaa_replication_status"
   Defaulted container "core" out of: core, noobaa-log-processor

   NooBaa_bucket_last_cycle_total_objects_num{bucket_name="test-buck1"} 15
   NooBaa_bucket_last_cycle_total_objects_num{bucket_name="test-buck2"} 25
   NooBaa_bucket_last_cycle_replicated_objects_num{bucket_name="test-buck1"} 15
   NooBaa_bucket_last_cycle_replicated_objects_num{bucket_name="test-buck2"} 25
   NooBaa_bucket_last_cycle_error_objects_num{bucket_name="test-buck1"} 0
   NooBaa_bucket_last_cycle_error_objects_num{bucket_name="test-buck2"} 0
   ```

2. Use the following syntax to view the metrics in the metrics section:

   ```
   metric_name {bucket_name='bucket_name'}
   ```

   For example:

   For `NooBaa_bucket_last_cycle_total_objects_num`

   ```
   NooBaa_bucket_last_cycle_total_objects_num {bucket_name='bubck1'}
   ```

   For `NooBaa_bucket_last_cycle_replicated_objects_num`

   ```
   NooBaa_bucket_last_cycle_replicated_objects_num {bucket_name='bubck1'}
   ```

   For `NooBaa_bucket_last_cycle_error_objects_num`

   ```
   NooBaa_bucket_last_cycle_error_objects_num {bucket_name='bubck1'}
   ```

Object Bucket Claim

An Object Bucket Claim can be used to request an S3 compatible bucket backend for your workloads. An object bucket claim creates a new bucket and an application account in NooBaa with permissions to the bucket, including a new access key and secret access key. The application account is allowed to access only a single bucket and can't create new buckets by default. Use this information to create, view, and delete object bucket claims.

Object Bucket Claim can be created in three ways: dynamically, using the command line interface, and using OpenShift Web Console.

- **Dynamic Object Bucket Claim**
  Similar to Persistent Volumes, you can add the details of the Object Bucket claim (OBC) to your application's YAML, and get the object service endpoint, access key, and secret access key available in a configuration map and secret. It is easy to read this information dynamically into environment variables of your application. Use this information to add the details of the Object Bucket claim (OBC) to your application's YAML.
- **Creating an Object Bucket Claim using the command line interface**
  Use this information to create an Object Bucket Claim using the command line interface.
- **Creating an Object Bucket Claim using the OpenShift Web Console**
  Use this information to create an Object Bucket Claim using the OpenShift Web Console.
- **Attaching an Object Bucket Claim to a deployment**
  Use this information to attach an Object Bucket Claim (OBC) to a deployment. Attaching to specific deployments can only be done after the OBCs are created.
- **Viewing object buckets using the OpenShift Web Console**
  You can view the details of object buckets created for Object Bucket Claims (OBCs) using the OpenShift Web Console.
- **Deleting Object Bucket Claims**
  When you no longer need the Object Bucket Claims, you can delete it from the namespace using Open Shift Web Console.

Dynamic Object Bucket Claim

Similar to Persistent Volumes, you can add the details of the Object Bucket claim (OBC) to your application's YAML, and get the object service endpoint, access key, and secret access key available in a configuration map and secret. It is easy to read this information dynamically into environment variables of your application. Use this information to add the details of the Object Bucket claim (OBC) to your application's YAML.

About this task

Note: The Multicloud Object Gateway endpoints uses self-signed certificates only if OpenShift uses self-signed certificates. Using signed certificates in OpenShift automatically replaces the Multicloud Object Gateway endpoints certificates with signed certificates. Get the certificate currently used by Multicloud Object Gateway by accessing the endpoint through the browser. For more information, see Accessing the Multicloud Object Gateway with your applications.

Procedure

1. Add the following line to your application YAML:
   These lines are the OBC itself.

   ```
   apiVersion: objectbucket.io/v1alpha1
   kind: ObjectBucketClaim
   metadata:
     name: <obc-name>
   spec:
     generateBucketName: <obc-bucket-name>
     storageClassName: openshift-storage.noobaa.io
   ```

   <obc-name>
   Use a unique OBC name.
   <obc-bucket-name>
   Use a unique bucket name for your OBC.

2. To automate the use of OBC, add more lines to the YAML file.
   For example:

   ```
   apiVersion: batch/v1
   kind: Job
   metadata:
     name: testjob
   spec:
   ```

```
  template:
    spec:
      restartPolicy: OnFailure
      containers:
        - image: <your application image>
          name: test
          env:
            - name: BUCKET_NAME
              valueFrom:
                configMapKeyRef:
                  name: <obc-name>
                  key: BUCKET_NAME
            - name: BUCKET_HOST
              valueFrom:
                configMapKeyRef:
                  name: <obc-name>
                  key: BUCKET_HOST
            - name: BUCKET_PORT
              valueFrom:
                configMapKeyRef:
                  name: <obc-name>
                  key: BUCKET_PORT
            - name: AWS_ACCESS_KEY_ID
              valueFrom:
                secretKeyRef:
                  name: <obc-name>
                  key: AWS_ACCESS_KEY_ID
            - name: AWS_SECRET_ACCESS_KEY
              valueFrom:
                secretKeyRef:
                  name: <obc-name>
                  key: AWS_SECRET_ACCESS_KEY
```

The example is the mapping between the bucket claim result, which is a configuration map with data and a secret with the credentials. This specific job claims the Object Bucket from NooBaa, which creates a bucket and an account.

<obc-name>
: Use your OBC name.

<your application image>
: Use your application image.

3. Apply the updated YAML file, where *<yaml.file>* is the name of your YAML file.

```
oc apply -f <yaml.file>
```

4. To view the new configuration map, run the following command, where *<obc-name>* is the name of your OBC.

```
oc get cm <obc-name> -o yaml
```

Expect the following environment variables in the output:

BUCKET_HOST
: Endpoint to use in the application.

BUCKET_HOST
: The port is related to the *BUCKET_HOST*. For example, if the `BUCKET_HOST` is `https://my.example.com`, and the `BUCKET_PORT` is 443, the endpoint for the object service would be `https://my.exaple.com:443`.

BUCKET_NAME
: Requested or generated bucket name.

AWS_ACCESS_KEY_ID
: Access key that is part of the credentials.

AWS_SECRET_ACCESS_KEY
: Secret access key that is part of the credentials.

Important: Retrieve the *AWS_ACCESS_KEY_ID* and *AWS_SECRET_ACCESS_KEY*. The names are used so that it is compatible with the AWS S3 API. You need to specify the keys while performing S3 operations, especially when you read, write or list from the Multicloud Object Gateway (MCG) bucket. The keys are encoded in Base64.
Decode the keys before using them, using the following command, where *<obc_name>* specifies the name of the object bucket claim:

```
oc get secret <obc_name> -o yaml
```

Creating an Object Bucket Claim using the command line interface

Use this information to create an Object Bucket Claim using the command line interface.

Before you begin

Download the Multicloud Object Gateway (MCG) command-line interface.

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
 yum install mcg
```

Note: Specify the appropriate architecture for enabling the repositories using the subscription manager.

- For IBM Power, use the following command:

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
```

- For IBM Z infrastructure, use the following command:

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
```

About this task

When creating an Object Bucket Claim (OBC) using the command-line interface, you get a configuration map and a Secret that together contain all the information your application needs to use the object storage service.

Procedure

1. Use the command-line interface to generate the details of a new bucket and credentials.

   ```
   odf noobaa obc create <obc-name> -n openshift-storage
   ```

   Replace *<obc-name>* with a unique OBC name, for example, **myappobc**.

   Additionally, you can use the **--app-namespace** option to specify the namespace where the OBC configuration map and secret are created, for example, **myapp-namespace**.

   For example:

   ```
   INFO[0001] ✅ Created: ObjectBucketClaim "test21obc"
   ```

2. Run the following command to view the OBC.

   ```
   oc get obc -n openshift-storage
   ```

   Example output:

   ```
   NAME         STORAGE-CLASS                 PHASE   AGE
   test21obc    openshift-storage.noobaa.io   Bound   38s
   ```

3. Run the following command to view the YAML file for the new OBC.

   ```
   oc get obc test21obc -o yaml -n openshift-storage
   ```

   Example YAML output:

   ```
   apiVersion: objectbucket.io/v1alpha1
   kind: ObjectBucketClaim
   metadata:
     creationTimestamp: "2019-10-24T13:30:07Z"
     finalizers:
     - objectbucket.io/finalizer
     generation: 2
     labels:
       app: noobaa
       bucket-provisioner: openshift-storage.noobaa.io-obc
       noobaa-domain: openshift-storage.noobaa.io
     name: test21obc
     namespace: openshift-storage
     resourceVersion: "40756"
     selfLink: /apis/objectbucket.io/v1alpha1/namespaces/openshift-storage/objectbucketclaims/test21obc
     uid: 64f04cba-f662-11e9-bc3c-0295250841af
   spec:
     ObjectBucketName: obc-openshift-storage-test21obc
     bucketName: test21obc-933348a6-e267-4f82-82f1-e59bf4fe3bb4
     generateBucketName: test21obc
     storageClassName: openshift-storage.noobaa.io
   status:
     phase: Bound
   ```

4. Inside of your **openshift-storage** namespace, you can find the configuration map and the secret to use this OBC. The CM and the secret have the same name as the OBC.

   ```
   oc get -n openshift-storage secret test21obc -o yaml
   ```

   For example:

   ```
   apiVersion: v1
   data:
     AWS_ACCESS_KEY_ID: c0M0R2xVanF3ODR3bHBkVW94cmY=
     AWS_SECRET_ACCESS_KEY: Wi9kcFluSWxHRzlWaFlzNk1hc0xma2JXcjM1MVhqa051SlBleXpmOQ==
   kind: Secret
   metadata:
     creationTimestamp: "2019-10-24T13:30:07Z"
     finalizers:
     - objectbucket.io/finalizer
     labels:
       app: noobaa
       bucket-provisioner: openshift-storage.noobaa.io-obc
       noobaa-domain: openshift-storage.noobaa.io
     name: test21obc
     namespace: openshift-storage
     ownerReferences:
     - apiVersion: objectbucket.io/v1alpha1
       blockOwnerDeletion: true
       controller: true
       kind: ObjectBucketClaim
       name: test21obc
       uid: 64f04cba-f662-11e9-bc3c-0295250841af
   ```

```
  resourceVersion: "40751"
  selfLink: /api/v1/namespaces/openshift-storage/secrets/test21obc
  uid: 65117c1c-f662-11e9-9094-0a5305de57bb
type: Opaque
```

The secret gives you S3 access credentials.

5. Run the following command to see the configuration map:

```
oc get -n openshift-storage cm test21obc -o yaml
```

For example:

```
apiVersion: v1
data:
  BUCKET_HOST: 10.0.171.35
  BUCKET_NAME: test21obc-933348a6-e267-4f82-82f1-e59bf4fe3bb4
  BUCKET_PORT: "31242"
  BUCKET_REGION: ""
  BUCKET_SUBREGION: ""
kind: ConfigMap
metadata:
  creationTimestamp: "2019-10-24T13:30:07Z"
  finalizers:
  - objectbucket.io/finalizer
  labels:
    app: noobaa
    bucket-provisioner: openshift-storage.noobaa.io-obc
    noobaa-domain: openshift-storage.noobaa.io
  name: test21obc
  namespace: openshift-storage
  ownerReferences:
  - apiVersion: objectbucket.io/v1alpha1
    blockOwnerDeletion: true
    controller: true
    kind: ObjectBucketClaim
    name: test21obc
    uid: 64f04cba-f662-11e9-bc3c-0295250841af
  resourceVersion: "40752"
  selfLink: /api/v1/namespaces/openshift-storage/configmaps/test21obc
  uid: 651c6501-f662-11e9-9094-0a5305de57bb
```

The configuration map contains the S3 endpoint information for your application.

Creating an Object Bucket Claim using the OpenShift Web Console

Use this information to create an Object Bucket Claim using the OpenShift Web Console.

Before you begin

Ensure you have the following:

- Administrative access to the OpenShift Web Console.
- In order for your applications to communicate with the OBC, you need to use the configmap and secret. For more information, see Dynamic Object Bucket Claim.

Procedure

From the OpenShift Web Console, go to Storage > Object Bucket Claims > Create Object Bucket Claim.

1. Enter a name for your object bucket claim.
2. Select the appropriate storage class based on your deployment, either internal or external, from the drop-down menu.
   - **Internal mode**
     The following storage classes, which were created after deployment, are available for use:

     `ocs-storagecluster-ceph-rgw`
     : Uses the Ceph Object Gateway (RGW).

     `openshift-storage.noobaa.io`
     : Uses the Multicloud Object Gateway (MCG).

   - **External mode**
     The following storage classes, which were created after deployment, are available for use:

     `ocs-external-storagecluster-ceph-rgw`
     : uses the RGW.

     `openshift-storage.noobaa.io`
     : Uses the MCG.

     Note: The RGW OBC storage class is only available with fresh installations of Fusion Data Foundation. It does not apply to clusters upgraded from previous OpenShift Data Foundation releases.
3. Click Create.
   After you create the OBC, you are redirected to its detail page.

Attaching an Object Bucket Claim to a deployment

Use this information to attach an Object Bucket Claim (OBC) to a deployment. Attaching to specific deployments can only be done after the OBCs are created.

Before you begin

Ensure you have Administrative access to the OpenShift Web Console.

Procedure

1. Go to Storage > Object Bucket Claim.
2. Click the Action menu next to Object Bucket Claim (OBC) you want to delete.
    a. From the drop-down menu, select Attach to deployment.
    b. Select the desired deployment from the Deployment Name list, then click Attach.

Viewing object buckets using the OpenShift Web Console

You can view the details of object buckets created for Object Bucket Claims (OBCs) using the OpenShift Web Console.

Before you begin

Ensure you have Administrative access to the OpenShift Web Console.

Procedure

1. Log into the Open Shift Web Console and go to Storage > Object Buckets.
   Alternatively, you can also navigate to the details page of a specific OBC, and click the Resource link to view the object buckets for that OBC.
2. Select the object bucket of which you want to see the details.
   After selected you are navigated to the Object Bucket Details page.

Deleting Object Bucket Claims

When you no longer need the Object Bucket Claims, you can delete it from the namespace using Open Shift Web Console.

Before you begin

Ensure you have Administrative access to the OpenShift Web Console.

Procedure

1. Go to Storage > Object Bucket Claims.
2. From next to the Object Bucket Claim (OBC) you want to delete, click Action Menu > Delete Object Bucket Claim
3. Click Delete.

Caching policy for object buckets

A cache bucket is a namespace bucket with a hub target and a cache target. The hub target is an S3 compatible large object storage bucket. The cache bucket is the local Multicloud Object Gateway bucket. You can create a cache bucket that caches an AWS bucket or an IBM COS bucket.

- **Creating an AWS cache bucket**
  Create an AWS cache bucket.
- **Creating an IBM COS cache bucket**
  Create an IBM COS cache bucket.

Creating an AWS cache bucket

Create an AWS cache bucket.

Before you begin

- A running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

Note: Specify the appropriate architecture for enabling the repositories using the subscription manager. In case of IBM Z infrastructure, use the following command:

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

About this task

This procedure describes creating a NamespaceStore. A NamespaceStore represents an underlying storage to be used as a read or write target for the data in the MCG namespace buckets.

Procedure

1. Create NamespaceStore in one of the following ways:
   - From the MCG command-line interface, run the following command:

     ```
     odf noobaa namespacestore create aws-s3 <namespacestore> --access-key <AWS ACCESS KEY> --secret-key <AWS SECRET ACCESS KEY> --target-bucket <bucket-name>
     ```

     namespacestore
     : Name of the namespacestore

     AWS ACCESS KEY and *AWS SECRET ACCESS KEY*
     : AWS access key ID and secret access key you created for this purpose.

     bucket-name
     : An existing AWS bucket name. This argument tells the MCG which bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.

   - Add storage resources by applying a YAML.
     - Create a secret with credentials, where *<namespacestore-secret-name>* is a unique name for the NamespaceStore secret.

       ```
       apiVersion: v1
       kind: Secret
       metadata:
         name: <namespacestore-secret-name>
       type: Opaque
       data:
         AWS_ACCESS_KEY_ID: <AWS ACCESS KEY ID ENCODED IN BASE64>
         AWS_SECRET_ACCESS_KEY: <AWS SECRET ACCESS KEY ENCODED IN BASE64>
       ```

       You must supply and encode your own AWS access key ID and secret access key using Base64, and use the results in place of *<AWS ACCESS KEY ID ENCODED IN BASE64>* and *<AWS SECRET ACCESS KEY ENCODED IN BASE64>*.

     - Apply the following YAML:

       ```
       apiVersion: noobaa.io/v1alpha1
       kind: NamespaceStore
       metadata:
         finalizers:
         - noobaa.io/finalizer
         labels:
           app: noobaa
         name: <namespacestore>
         namespace: openshift-storage
       spec:
         awsS3:
           secret:
             name: <namespacestore-secret-name>
             namespace: <namespace-secret>
           targetBucket: <target-bucket>
         type: aws-s3
       ```

       namespacestore
       : A unique name for the NamespaceStore secret.

       namespacestore-secret-name
       : A name for the NamespaceStore secret created in the previous step.

       namespace-secret
       : Namespace used to create the secret in the previous step.

       target-bucket
       : the AWS S3 bucket you created for the NamespaceStore.

2. Run the following command to create a bucket class:

   ```
   odf noobaa bucketclass create namespace-bucketclass cache <my-cache-bucket-class> --backingstores <backing-store> --hub-resource <namespacestore>
   ```

   my-cache-bucket-class
   : A unique bucket class name.

   backing-store
   : Name of the relevant backing store. You can also list more than one backing stores separated by commas.

   namespacestore
   : Name of the NamespaceStore created in the previous step.

3. Run the following command to create a bucket using an Object Bucket Claim (OBC) resource that uses the bucket class defined in the previous step.

```
odf noobaa obc create <my-bucket-claim> my-app --bucketclass <custom-bucket-class>
```

my-bucket-claim
: A unique object bucket claim name.

custom-bucket-class
: Name of the bucket class created in the previous step.

Creating an IBM COS cache bucket

Create an IBM COS cache bucket.

Before you begin

- A running Fusion Data Foundation Platform.
- Download the MCG command-line interface for easier management.

  ```
  subscription-manager repos --enable=rh-odf-4-for-rhel-8-x86_64-rpms
  ```

  ```
  yum install mcg
  ```

  Note: Specify the appropriate architecture for enabling the repositories using the subscription manager.
  - For IBM Power, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
    ```

  - For IBM Z infrastructure, use the following command:

    ```
    subscription-manager repos --enable=rh-odf-4-for-rhel-8-s390x-rpms
    ```

- Alternatively, you can install the MCG package from the RPMs found at Download Red Hat OpenShift Data Foundation page.

  Note: Choose the correct Product Variant according to your architecture.

About this task

This procedure describes creating a NamespaceStore. A NamespaceStore represents an underlying storage to be used as a read or write target for the data in the MCG namespace buckets.

Procedure

1. Create NamespaceStore in one of the following ways:
   - From the MCG command-line interface, run the following command:

     ```
     odf noobaa namespacestore create ibm-cos <namespacestore> --endpoint <IBM COS ENDPOINT> --access-key <IBM ACCESS KEY> --
     ```

     namespacestore
     : Name of the NamespaceStore

     IBM COS ENDPOINT
     : An appropriate regional endpoint that corresponds to the location of the existing IBM bucket.

     IBM ACCESS KEY and *IBM SECRET ACCESS KEY*
     : IBM access key ID and secret access key you created for this purpose.

     bucket-name
     : An existing IBM bucket name. This argument tells the MCG which bucket to use as a target bucket for its backing store, and subsequently, data storage and administration.

   - Add storage resources by applying a YAML.
     - Create a secret with credentials, where *<namespacestore-secret-name>* is a unique name for the NamespaceStore secret.

       ```
       apiVersion: v1
       kind: Secret
       metadata:
         name: <namespacestore-secret-name>
       type: Opaque
       data:
         IBM_COS_ACCESS_KEY_ID: <IBM COS ACCESS KEY ID ENCODED IN BASE64>
         IBM_COS_SECRET_ACCESS_KEY: <IBM COS SECRET ACCESS KEY ENCODED IN BASE64>
       ```

       You must supply and encode your own IBM COS access key ID and secret access key using Base64, and use the results in place of *<IBM COS ACCESS KEY ID ENCODED IN BASE64>* and *<IBM COS SECRET ACCESS KEY ENCODED IN BASE64>*.

     - Apply the following YAML:

       ```
       apiVersion: noobaa.io/v1alpha1
       kind: NamespaceStore
       metadata:
         finalizers:
         - noobaa.io/finalizer
         labels:
           app: noobaa
         name: <namespacestore>
         namespace: openshift-storage
       spec:
       ```

```
  s3Compatible:
    endpoint: <IBM COS ENDPOINT>
    secret:
      name: <backingstore-secret-name>
      namespace: <namespace-secret>
    signatureVersion: v2
    targetBucket: <target-bucket>
  type: ibm-cos
```

namespacestore
: A unique name for the NamespaceStore secret.

IBM COS ENDPOINT
: An appropriate regional endpoint that corresponds to the location of the existing IBM bucket.

backingstore-secret-name
: A name for the backing store secret created in the previous step.

namespace-secret
: Namespace used to create the secret in the previous step.

target-bucket
: the IBM COS bucket you created for the NamespaceStore.

2. Run the following command to create a bucket class:

```
odf noobaa bucketclass create namespace-bucketclass cache <my-bucket-class> --backingstores <backing-store> --hub-resource <namespacestore>
```

my-bucket-class
: A unique bucket class name.

backing-store
: Name of the relevant backing store. You can also list more than one backing stores separated by commas.

namespacestore
: Name of the NamespaceStore created in the previous step.

3. Run the following command to create a bucket using an Object Bucket Claim (OBC) resource that uses the bucket class defined in the previous step.

```
odf noobaa obc create <my-bucket-claim> my-app --bucketclass <custom-bucket-class>
```

my-bucket-claim
: A unique object bucket claim name.

custom-bucket-class
: Name of the bucket class created in the previous step.

Lifecycle bucket configuration in Multicloud Object Gateway

Multicloud Object Gateway (MCG) lifecycle provides a way to reduce storage costs due to accumulated data objects.

Deletion of expired objects is a simplified way that enables handling of unused data. Data expiration is a part of Amazon Web Services (AWS) lifecycle management and sets an expiration date for automatic deletion. The minimal time resolution of the lifecycle expiration is one day. For more information, see Expiring objects.

AWS S3 API is used to configure lifecycle bucket in MCG. For information about the data bucket APIs and their support level, see Support of Multicloud Object Gateway data bucket APIs.

There are a few limitations with the expiratation rule API for MCG in comparison with AWS:

- `ExpiredObjectDeleteMarker` is accepted but it is not processed.
- No option to define specific non-current version's expiration conditions

The expiration rule API for MCG doesn't support transition rules like AWS does for both current and noncurrent objects.

You can configure the bucket lifecycle by using the MCG object browser. For more information, see Creating new lifecycle rules by using the object browser.

Scaling Multicloud Object Gateway performance

The Multicloud Object Gateway (MCG) performance may vary from one environment to another. In some cases, specific applications require faster performance which can be addressed by scaling S3 endpoints.

The MCG resource pool is a group of NooBaa daemon containers that provide two types of services enabled by default:

- Storage service
- S3 endpoint service

S3 endpoint service

The S3 endpoint is a service that every Multicloud Object Gateway (MCG) provides by default that handles the heavy lifting data digestion in the MCG. The endpoint service handles the inline data chunking, deduplication, compression, and encryption, and it accepts data placement instructions from the MCG.

- **Automatic scaling of MultiCloud Object Gateway endpoints**
  The number of MultiCloud Object Gateway (MCG) endpoints scale automatically when the load on the MCG S3 service increases or decreases.
- **Autoscaling RGW in Fusion Data Foundation using HPA and KEDA**
  Enable autoscaling for RGW (RADOS Gateway) in Fusion Data Foundation by integrating with OpenShift Horizontal Pod Autoscaler (HPA) with KEDA (Kubernetes Event-Driven Autoscaling).

- **Scaling the Multicloud Object Gateway with storage nodes**
  A storage node in the MCG is a NooBaa daemon container attached to one or more Persistent Volumes (PVs) and used for local object service data storage. NooBaa daemons can be deployed on Kubernetes nodes. This can be done by creating a Kubernetes pool consisting of `StatefulSet` pods.
- **Increasing CPU and memory for PV pool resources**
  Multicloud Object Gateway (MCG) default configuration supports low resource consumption. However, when you need to increase CPU and memory to accommodate specific workloads and to increase MCG performance for the workloads, it is possible to configure the required values for CPU and memory in the OpenShift Web Console.

Automatic scaling of MultiCloud Object Gateway endpoints

The number of MultiCloud Object Gateway (MCG) endpoints scale automatically when the load on the MCG S3 service increases or decreases.

Fusion Data Foundation clusters are deployed with one active MCG endpoint. Each MCG endpoint pod is configured by default with 1 CPU and 2Gi memory request, with limits matching the request. When the CPU load on the endpoint crosses over an 80% usage threshold for a consistent period of time, a second endpoint is deployed lowering the load on the first endpoint. When the average CPU load on both endpoints falls below the 80% threshold for a consistent period of time, one of the endpoints is deleted. This feature improves performance and serviceability of the MCG.

Autoscaling RGW in Fusion Data Foundation using HPA and KEDA

Enable autoscaling for RGW (RADOS Gateway) in Fusion Data Foundation by integrating with OpenShift Horizontal Pod Autoscaler (HPA) with KEDA (Kubernetes Event-Driven Autoscaling).

Before you begin

- All resources described in this procedure (Roles, RoleBindings, TriggerAuthentication, ScaledObject, and so on) must be created in the `openshift-storage` namespace.
- Custom Metrics Autoscaler Operator version must be 2.6 or later.

Procedure

1. Install the Custom Metrics Autoscaler Operator.
   a. Log in to the Red Hat® OpenShift® Container Platform web console.
   b. Click Ecosystem > Software Catalog.
   c. Scroll or type Custom Metrics Autoscaler into the Filter by keyword box to find the Custom Metrics Autoscaler Operator.
   d. Click Install.
2. Create a Custom Metrics Autoscaler controller and verify whether the following pods are running in the `openshift-keda` namespace:

```
oc get pods -n openshift-keda
NAME                                                READY   STATUS    RESTARTS   AGE
custom-metrics-autoscaler-operator-6cb698bb4d-nnpmj   1/1     Running   0          103m
keda-admission-c6d879546-twprp                      1/1     Running   0          103m
keda-metrics-apiserver-68788fbd9b-h7l5n             1/1     Running   0          103m
keda-operator-664dc57859-mb49f                      1/1     Running   0          20m
```

3. Configure Custom Metrics Autoscaler with the Thanos Query Service.
   Prometheus scaler that is supported by Custom Metrics Autoscaler collects metrics through the Thanos Query service provided by OpenShift Monitoring. Because Thanos Query requires authentication, a ServiceAccount token must be used for bearer authentication.

   a. Create a service account.
      Command example:

```
oc create serviceaccount thanos -n openshift-storage
```

   b. Create a secret to generate the token.

```
apiVersion: v1
kind: Secret
metadata:
  name: thanos-token-secret
  namespace: openshift-storage
  annotations:
    kubernetes.io/service-account.name: "thanos"
type: kubernetes.io/service-account-token
```

      Command example:

```
oc apply -f secret.yaml
```

   c. Add the `cluster-monitoring-view` cluster role to the created Service Account in the `openshift-storage` namespace.
      Command example:

```
oc adm policy add-cluster-role-to-user cluster-monitoring-view -z thanos
```

   d. Create a `triggerAuthentication` CRD for the bearer authentication using the secret name.

```
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: keda-trigger-auth-prometheus
  namespace: openshift-storage
```

```
spec:
  secretTargetRef:
  - parameter: bearerToken
    name: thanos-token-secret
    key: token
  - parameter: ca
    name: thanos-token-secret
    key: ca.crt
```

4. Enable RGW autoscaling.
   Command example:

   ```
   oc annotate storagecluster ocs-storagecluster ocs.openshift.io/enable-rgw-autoscale="true" -n openshift-storage
   ```

   This behavior takes precedence over manual scaling. You can revert this change by performing the following command:

   ```
   oc annotate storagecluster ocs-storagecluster ocs.openshift.io/enable-rgw-autoscale- -n openshift-storage
   ```

5. Create ScaledObject for autoscaling RGW.
   Autoscaling using the HPA feature of OpenShift triggers scaling based on custom metrics. The prometheus metrics related to RGW exported by the Ceph manager is used. The following example uses `ceph_rgw_put_obj_ops` metrics and scale up to five pods.

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: rgw-scale-namespace
  namespace: openshift-storage
spec:
  scaleTargetRef:
    apiVersion: ceph.rook.io/v1
    kind: CephObjectStore
    name: ocs-storagecluster-cephobjectstore
    namespace: openshift-storage
  minReplicaCount: 1
  maxReplicaCount: 5
  triggers:
    - type: prometheus
      metadata:
        serverAddress: https://thanos-querier.openshift-monitoring.svc.cluster.local:9091
        metricName: ceph_rgw_put_obj_ops
        query: |
          sum(rate(ceph_rgw_op_put_obj_ops[2m]))
        threshold: "1"
        authModes: "bearer"
        namespace: openshift-storage
      authenticationRef:
        name: keda-trigger-auth-prometheus
```

Scaling the Multicloud Object Gateway with storage nodes

A storage node in the MCG is a NooBaa daemon container attached to one or more Persistent Volumes (PVs) and used for local object service data storage. NooBaa daemons can be deployed on Kubernetes nodes. This can be done by creating a Kubernetes pool consisting of `StatefulSet` pods.

Before you begin

Ensure you have a running Fusion Data Foundation cluster on OpenShift Container Platform with access to the Multicloud Object Gateway (MCG).

Procedure

1. Log in to the OpenShift Web Console.
2. From the MCG user interface, go to Overview > Add Storage Resources.
3. Click **Deploy Kubernetes Pool**.
4. In the Create Pool step, create the target pool for the future installed nodes.
5. In the Configure step, configure the number of requested pods and the size of each PV.
   For each new pod, one PV is to be created
6. In the Review step, you can find the details of the new pool and select the deployment method you wish to use: local or external deployment.
   If local deployment is selected, the Kubernetes nodes deploy within the cluster. If external deployment is selected, a YAML file is provided to run externally.

What to do next

All nodes are assigned to the pool you chose in the first step. To verify, go to Resources > Storage resources > Resource name.

Increasing CPU and memory for PV pool resources

Multicloud Object Gateway (MCG) default configuration supports low resource consumption. However, when you need to increase CPU and memory to accommodate specific workloads and to increase MCG performance for the workloads, it is possible to configure the required values for CPU and memory in the OpenShift Web Console.

Procedure

1. From the OpenShift Web Console, go to Storage > Object storage > Backingstore tab.
2. Select the new backingstore.
3. Click Edit PV pool resources.
4. From the edit window, edit the following values, based on your requirements: based on the requirement.
   Mem, CPU, and Vol size
5. Click **Save**.

What to do next

To verify, check the resource values of the PV pool pods.

Accessing the RADOS Object Gateway S3 endpoint

Users can access the RADOS Object Gateway (RGW) endpoint directly. The RGW route is created by default and is named **`rook-ceph-rgw-ocs-storagecluster-cephobjectstore`**.

Using TLS certificates for applications accessing RGW

Fetch TLS certificates stored as Kubernetes secrets and configure applications to use them when accessing the RADOS Object Gateway (RGW) over a secure connection.

Before you begin

A running Fusion Data Foundation cluster.

About this task

Most of the S3 applications require TLS certificate in the forms such as an option included in the Deployment configuration file, passed as a file in the request, or stored in **`/etc/pki`** paths.

TLS certificates for RADOS Object Gateway (RGW) are stored as Kubernetes secret and you need to fetch the details from the secret.

Procedure

- For internal RGW server
  Get the TLS certificate and key from the kubernetes secret:

  ```
  oc get secrets/<secret_name> --template={{.data.tls.crt}} | base64 -d
  ```

  ```
  oc get secrets/<secret_name> --template={{.data.tls.key}} | base64 -d
  ```

  <secret_name>

  The default kubernetes secret name is **`<objectstore_name>-cos-ceph-rgw-tls-cert`**. Specify the name of the object store.

- For external RGW server
  Get the TLS certificate from the kubernetes secret:

  ```
  oc get secrets/<secret_name> --template={{.data.cert}} | base64 -d
  ```

  <secret_name>

  The default kubernetes secret name is **`ceph-rgw-tls-cert`** and it is an opaque type of secret. The key value for storing the TLS certificates is **`cert`**.

- **Accessing external RGW server**
  This section describes the two ways of accessing external RGW server

Accessing external RGW server

This section describes the two ways of accessing external RGW server

The S3 credentials such as **`AccessKey`** or **`Secret Key`** is stored in the secret generated by the Object Bucket Claim (OBC) creation and you can fetch the same by using the following commands:

Accessing External RGW server using Object Bucket Claims

```
oc get secret <object bucket claim name> -o jsonpath='{.data.AWS_SECRET_ACCESS_KEY}' | base64 --decode
oc get secret <object bucket claim name> -o jsonpath='{.data.AWS_ACCESS_KEY_ID}' | base64 --decode
```

Similarly, you can fetch the endpoint details from the **`configmap`** of OBC:

```
oc get cm <object bucket claim name> -o jsonpath='{.data.BUCKET_HOST}'
oc get cm <object bucket claim name> -o jsonpath='{.data.BUCKET_PORT}'
oc get cm <object bucket claim name> -o jsonpath='{.data.BUCKET_NAME}'
```

Accessing External RGW server using the `Ceph Object Store User` CR

You can fetch the S3 Credentials and endpoint details from the secret generated as part of the **`Ceph Object Store User`** CR:

```
oc get secret rook-ceph-object-user-<object-store-cr-name>-<object-user-cr-name> -o jsonpath='{.data.AccessKey}' | base64 --decode
oc get secret rook-ceph-object-user-<object-store-cr-name>-<object-user-cr-name> -o jsonpath='{.data.SecretKey}' | base64 --decode
oc get secret rook-ceph-object-user-<object-store-cr-name>-<object-user-cr-name> -o jsonpath='{.data.Endpoint}' | base64 --decode
```

Important:
For both the access mechanisms, you can either request for new certificates from the administrator or reuse the certificates from the Kubernetes secret, **`ceph-rgw-tls-cert`**.

Using the security token service of Multicloud Object Gateway to assume the role of another user

Multicloud Object Gateway (MCG) supports for a security token service (STS) similar to the one provided by Amazon Web Services. This STS of the MCG authorizes a user to assume the role of another user.

About this task

To authorize other users to assume the role of a certain user, you need to assign a role configuration to the user. You can manage the configuration of roles that use the MCG command-line interface (CLI) tool.

The following example shows the role configuration that allows two MCG users (*assumer@mcg.test* and *assumer2@mcg.test*) to assume a certain user's role:

```
'{"role_name": "AllowTwoAssumers", "assume_role_policy": {"version": "2012-10-17", "statement": [ {"action": ["sts:AssumeRole"], "effect": "allow", "principal": ["assumer@mcg.test", "assumer2@mcg.test"]}]}}'
```

Procedure

1. Assign the role configuration by using the MCG CLI tool.

   ```
   mcg sts assign-role --email <assumed user's username> --role_config '{"role_name": "AllowTwoAssumers", "assume_role_policy": {"version": "2012-10-17", "statement": [ {"action": ["sts:AssumeRole"], "effect": "allow", "principal": ["assumer@mcg.test", "assumer2@mcg.test"]}]}}'
   ```

2. Collect the following information:
   - The access key ID and secret access key of the assumer (the user who assumes the role).
   - The MCG STS endpoint, which can be retrieved by using the command:

     ```
     oc -n openshift-storage get route
     ```

   - The access key ID of the assumed user.
   - The value of the *role_name* in your role configuration.
   - A name of your choice for the role session.
3. Assign the configuration role to the appropriate user.

   ```
   AWS_ACCESS_KEY_ID=<aws-access-key-id> AWS_SECRET_ACCESS_KEY=<aws-secret-access-key1> aws --endpoint-url <mcg-sts-endpoint> sts assume-role --role-arn arn:aws:sts::<assumed-user-access-key-id>:role/<role-name> --role-session-name <role-session-name>
   ```

   Note: Adding --no-verify-ssl might be necessary depending on the configuration of the cluster.
   The output contains the access key ID, secret access key, and session token that can be used for running actions while assuming the other user's role.

   You can use the credentials generated after the assume role steps, as shown in the following example:

   ```
   AWS_ACCESS_KEY_ID=<aws-access-key-id> AWS_SECRET_ACCESS_KEY=<aws-secret-access-key1> AWS_SESSION_TOKEN=<session token> aws --endpoint-url <mcg-s3-endpoint> s3 ls
   ```

Replacing nodes

Explains how to safely replace a node in a Fusion Data Foundation cluster.

Fusion Data Foundation node replacement can be performed proactively for an operational node and reactively for a failed node.

Amazon Web Services (AWS)

- User-provisioned infrastructure
- Installer-provisioned infrastructure

VMware

- User-provisioned infrastructure
- Installer-provisioned infrastructure

Microsoft Azure

- Installer-provisioned infrastructure

Google Cloud

- Installer-provisioned infrastructure

Local storage devices

- Bare metal
- VMware
- IBM Power

Note: For replacing your storage nodes in external mode, see IBM Storage Ceph documentation .

- **Replacing nodes on Fusion Data Foundation using dynamic devices**
  Use this information to learn how to replace nodes on Fusion Data Foundation using dynamic devices.
- **Replacing nodes on Fusion Data Foundation using local storage devices**
  Use this information to learn how to replace nodes on Fusion Data Foundation using local storage devices.

Replacing nodes on Fusion Data Foundation using dynamic devices

Use this information to learn how to replace nodes on Fusion Data Foundation using dynamic devices.

- **Fusion Data Foundation deployed on AWS**
  Use this information for replacing an operational and failed AWS node on user-provisioned and installer-provisioned infrastructures.
- **Fusion Data Foundation deployed on VMware**
  Use this information for replacing an operational and failed VMware node on user-provisioned and installer-provisioned infrastructures.
- **Fusion Data Foundation deployed on Microsoft Azure**
  Use this information for replacing an operational and failed Microsoft Azure node on installer-provisioned infrastructure.
- **Fusion Data Foundation deployed on Google Cloud Platform**
  Use this information for replacing an operational and failed Google Cloud Platform (GCP) node on installer-provisioned infrastructures.

Fusion Data Foundation deployed on AWS

Use this information for replacing an operational and failed AWS node on user-provisioned and installer-provisioned infrastructures.

- **Replacing an operational AWS node on user-provisioned infrastructure**
  Use this information to replace an operational AWS node on a user-provisioned infrastructure.
- **Replacing an operational AWS node on installer-provisioned infrastructure**
  Use this information to replace an operational AWS node on an installer-provisioned infrastructure.
- **Replacing a failed AWS node on user-provisioned infrastructure**
  Use this information to replace failed AWS node on a user-provisioned infrastructure.
- **Replacing a failed AWS node on installer-provisioned infrastructure**
  Use this information to replace a failed AWS node on an installer-provisioned infrastructure.

Replacing an operational AWS node on user-provisioned infrastructure

Use this information to replace an operational AWS node on a user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node that you need to replace.
2. Mark the node as unscheduable, where *<node_name>* specifies the name of node that you need to replace.

   ```
   oc adm cordon <node_name>
   ```

3. Drain the node.

   ```
   oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
   ```

   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
4. Delete the node.

   ```
   oc delete nodes <node_name>
   ```

5. Create a new Amazon Web Service (AWS) machine instance with the required infrastructure.
   For more information, see Platform requirements.
6. Create a new OpenShift Container Platform node using the new AWS machine instance.

7. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in a *Pending* state.

   ```
   oc get csr
   ```

8. Approve all the required OpenShift Container Platform CSRs for the new node, where *<certificate_name>* specifies the name of the CSR.

   ```
   oc adm certificate approve <certificate_name>
   ```

9. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
10. Apply the Fusion Data Foundation label to the new node using one of the following steps:

    From the user interface
    a. Go to Action Menu > Edit Labels > .
    b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

    From the command-line interface
    Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

    ```
    oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
    ```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      oc debug node/<node_name>
      ```

      ```
      chroot /host
      ```

   b. Display the list of available block devices:, using the **lsblk** command.
      Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Replacing an operational AWS node on installer-provisioned infrastructure

Use this information to replace an operational AWS node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the node that you need to replace.
   Take a note of its Machine Name.
3. Mark the node as unschedulable, using the following command, where *<node_name>* specifies the name of the node that you need to replace.

   ```
   oc adm cordon <node_name>
   ```

4. Drain the node.

   ```
   oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
   ```

   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
5. Click Compute > Machines and search for the required machine.
6. For the required machine, click Action menu > Delete Machine.
7. Click **Delete** to confirm that the machine is deleted.
   A new machine is automatically created.
8. Wait for the new machine to start and transition into the *Running* state.
   Important: This activity might take at least 5 - 10 minutes or more.
9. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
10. Apply the Fusion Data Foundation label to the new node using one of the following steps:

    From the user interface
    a. Go to Action Menu > Edit Labels > .
    b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

    From the command-line interface
    Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

```
oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      oc debug node/<node_name>
      ```

      ```
      chroot /host
      ```

   b. Display the list of available block devices:, using the **lsblk** command.
      Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Replacing a failed AWS node on user-provisioned infrastructure

Use this information to replace failed AWS node on a user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the Amazon Web Service (AWS) machine instance of the node that you need to replace.
2. Log in to AWS, and terminate the AWS machine instance that you identified.
3. Create a new AWS machine instance with the required infrastructure.
   For more information, see Platform requirements.
4. Create a new OpenShift Container Platform node using the new AWS machine instance.
5. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in *Pending* state.

   ```
   oc get csr
   ```

6. Approve all the required OpenShift Container Platform CSRs for the new node, where *<certificate_name>* specifies the name of the CSR.

   ```
   oc adm certificate approve <certificate_name>
   ```

7. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
8. Apply the Fusion Data Foundation label to the new node using one of the following steps:

   From the user interface
   a. Go to Action Menu > Edit Labels > .
   b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

   From the command-line interface
   Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

   ```
   oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   oc debug node/<node_name>
   ```

   ```
   chroot /host
   ```

   b. Display the list of available block devices:, using the **lsblk** command.
   Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Replacing a failed AWS node on installer-provisioned infrastructure

Use this information to replace a failed AWS node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the faulty node that you need to replace and click on its Machine Name.
3. Go to Actions > Edit Annotations and click Add More.
4. Add `machine.openshift.io/exclude-node-draining`, and click Save.
5. Go to Action menu > Delete Machine and click Delete.
   A new machine is automatically created, wait for new machine to start.
   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
6. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
7. Apply the Fusion Data Foundation label to the new node using one of the following steps:

   From the user interface
   a. Go to Action Menu > Edit Labels > .
   b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

   From the command-line interface
   Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

   ```
   oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

8. Optional: If the failed Amazon Web Service (AWS) instance is not removed automatically, terminate the instance from the AWS console.

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   oc debug node/<node_name>
   ```

   ```
   chroot /host
   ```

   b. Display the list of available block devices:, using the **lsblk** command.
   Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Fusion Data Foundation deployed on VMware

Use this information for replacing an operational and failed VMware node on user-provisioned and installer-provisioned infrastructures.

- **Replacing an operational VMware node on user-provisioned infrastructure**
  Use this information to replace an operational VMware node on a user-provisioned infrastructure.

- **Replacing an operational VMware node on installer-provisioned infrastructure**
  Use this information to replace an operational VMware node on an installer-provisioned infrastructure.
- **Replacing a failed VMware node on user-provisioned infrastructure**
  Use this information to replace a failed VMware node on a user-provisioned infrastructure.
- **Replacing a failed VMware node on installer-provisioned infrastructure**
  Use this information to replace a failed VMware node on an installer-provisioned infrastructure.

Replacing an operational VMware node on user-provisioned infrastructure

Use this information to replace an operational VMware node on a user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node and its Virtual Machine (VM) that you need replace.
2. Mark the node as unscheduable, where *<node_name>* specifies the name of the node that needs to be replaced.

   ```
   oc adm cordon <node_name>
   ```

3. Drain the node.

   ```
   oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
   ```

   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when when you label the new node, and it is functional.
4. Delete the node.

   ```
   oc delete nodes <node_name>
   ```

5. Log in to VMware vSphere, and terminate the VM that you identified.
   Important: Delete the VM only from the inventory and not from the disk.
6. Create a new VM on VMware vSphere with the required infrastructure.
   For more information, see Platform requirements.
7. Create a new OpenShift Container Platform worker node using the new VM.
8. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in *Pending* state.

   ```
   oc get csr
   ```

9. Approve all the required OpenShift Container Platform CSRs for the new node, where *<certificate_name>* specifies the name of the CSR.

   ```
   oc adm certificate approve <certificate_name>
   ```

10. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
11. Apply the Fusion Data Foundation label to the new node using one of the following steps:

    From the user interface
    a. Go to Action Menu > Edit Labels > .
    b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

    From the command-line interface
    Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

    ```
    oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
    ```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   oc debug node/<node_name>
   ```

   ```
   chroot /host
   ```

b. Display the list of available block devices:, using the **lsblk** command.
Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact IBM Support.

Replacing an operational VMware node on installer-provisioned infrastructure

Use this information to replace an operational VMware node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the faulty node that you need to replace.
Take note of its Machine Name.
3. Mark the node as unscheduable, where *<node_name>* specifies the name of node that you need to replace.
```
oc adm cordon
<node_name>
```
4. Drain the node.
```
oc adm drain <node_name>
--force --delete-emptydir-data=true --ignore-daemonsets
```
Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
5. Click Compute > Machines and search for the required machine.
6. For the required machine, click Action menu > Delete Machine.
7. Click **Delete** to confirm that the machine is deleted.
A new machine is automatically created.
8. Wait for the new machine to start and transition into *Running* state.
Important: This activity might take at least 5 - 10 minutes or more.
9. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
10. Apply the Fusion Data Foundation label to the new node using one of the following steps:

From the user interface
a. Go to Action Menu > Edit Labels > .
b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

From the command-line interface
Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

```
oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

```
oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
For each of the new nodes identified in the previous step, do the following:

a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
oc debug node/<node_name>
```

```
chroot /host
```

b. Display the list of available block devices:, using the **lsblk** command.
Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact IBM Support.

Replacing a failed VMware node on user-provisioned infrastructure

Use this information to replace a failed VMware node on a user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node and its Virtual Machine (VM) that you need replace.
2. Delete the node, where *<node_name>* specifies the name of the node that needs to be replaced.

   ```
   oc delete nodes <node_name>
   ```

3. Log in to VMware vSphere, and terminate the VM that you identified.
   Important: Delete the VM only from the inventory and not from the disk.
4. Create a new VM on VMware vSphere with the required infrastructure.
   For more information, see Platform requirements.
5. Create a new OpenShift Container Platform worker node using the new VM.
6. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in *Pending* state.

   ```
   oc get csr
   ```

7. Approve all the required OpenShift Container Platform CSRs for the new node, where *<certificate_name>* specifies the name of the CSR.

   ```
   oc adm certificate approve <certificate_name>
   ```

8. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
9. Apply the Fusion Data Foundation label to the new node using one of the following steps:

   From the user interface
   a. Go to Action Menu > Edit Labels > .
   b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

   From the command-line interface
   Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

   ```
   oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      oc debug node/<node_name>
      ```

      ```
      chroot /host
      ```

   b. Display the list of available block devices:, using the **lsblk** command.
      Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Replacing a failed VMware node on installer-provisioned infrastructure

Use this information to replace a failed VMware node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the faulty node that you need to replace and click on its Machine Name.
3. Go to Actions > Edit Annotations and click Add More.
4. Add `machine.openshift.io/exclude-node-draining`, and click Save.
5. Go to Action menu > Delete Machine and click Delete.
   A new machine is automatically created, wait for new machine to start.
   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
6. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
7. Apply the Fusion Data Foundation label to the new node using one of the following steps:

   From the user interface
   a. Go to Action Menu > Edit Labels > .
   b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

From the command-line interface

Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

```
oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      oc debug node/<node_name>
      ```

      ```
      chroot /host
      ```

   b. Display the list of available block devices:, using the **lsblk** command.
      Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Fusion Data Foundation deployed on Microsoft Azure

Use this information for replacing an operational and failed Microsoft Azure node on installer-provisioned infrastructure.

- **Replacing operational nodes on Azure installer-provisioned infrastructure**
  Use this information to replace an operational Azure node on an installer-provisioned infrastructure.
- **Replacing failed nodes on Azure installer-provisioned infrastructure**
  Use this information to replace a failed Azure node on an installer-provisioned infrastructure.

Replacing operational nodes on Azure installer-provisioned infrastructure

Use this information to replace an operational Azure node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the faulty node that you need to replace.
   Take note of its Machine Name.
3. Mark the node as unschedulable, where *<node_name>* specifies the name of node that you need to replace.

   ```
   oc adm cordon <node_name>
   ```

4. Drain the node.

   ```
   oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
   ```

   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
5. Click Compute > Machines and search for the required machine.
6. For the required machine, click Action menu > Delete Machine.
7. Click **Delete** to confirm that the machine is deleted.
   A new machine is automatically created.
8. Wait for the new machine to start and transition into *Running* state.
   Important: This activity might take at least 5 - 10 minutes or more.
9. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
10. Apply the Fusion Data Foundation label to the new node using one of the following steps:

    From the user interface

    a. Go to Action Menu > Edit Labels > .
    b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

    From the command-line interface

    Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

    ```
    oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
    ```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      oc debug node/<node_name>
      ```

      ```
      chroot /host
      ```

   b. Display the list of available block devices:, using the **lsblk** command.
      Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Replacing failed nodes on Azure installer-provisioned infrastructure

Use this information to replace a failed Azure node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the faulty node that you need to replace and click on its Machine Name.
3. Go to Actions > Edit Annotations and click Add More.
4. Add `machine.openshift.io/exclude-node-draining`, and click Save.
5. Go to Action menu > Delete Machine and click Delete.
   A new machine is automatically created, wait for new machine to start.
   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
6. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
7. Apply the Fusion Data Foundation label to the new node using one of the following steps:

   From the user interface
   a. Go to Action Menu > Edit Labels > .
   b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

   From the command-line interface
   Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

   ```
   oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

8. Optional: If the failed Azure instance is not removed automatically, terminate the instance from the Azure console.

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      oc debug node/<node_name>
      ```

      ```
      chroot /host
      ```

b. Display the list of available block devices:, using the **lsblk** command.
Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact IBM Support.

Fusion Data Foundation deployed on Google Cloud Platform

Use this information for replacing an operational and failed Google Cloud Platform (GCP) node on installer-provisioned infrastructures.

- **Replacing an operational node on Google Cloud Platform installer-provisioned infrastructure**
  Use this information to replace an operational GCP node on an installer-provisioned infrastructure.
- **Replacing a failed Google Cloud Platform node on installer-provisioned infrastructure**
  Use this information to replace a failed GCP node on an installer-provisioned infrastructure.

Replacing an operational node on Google Cloud Platform installer-provisioned infrastructure

Use this information to replace an operational GCP node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the node that you need to replace.
   Take a note of its Machine Name.
3. Mark the node as unschedulable, using the following command, where *<node_name>* specifies the name of the node that you need to replace.

   ```
   oc adm cordon <node_name>
   ```

4. Drain the node.

   ```
   oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
   ```

   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
5. Click Compute > Machines and search for the required machine.
6. For the required machine, click Action menu > Delete Machine.
7. Click **Delete** to confirm that the machine is deleted.
   A new machine is automatically created.
8. Wait for the new machine to start and transition into the *Running* state.
   Important: This activity might take at least 5 - 10 minutes or more.
9. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
10. Apply the Fusion Data Foundation label to the new node using one of the following steps:

    From the user interface
    a. Go to Action Menu > Edit Labels > .
    b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

    From the command-line interface
    Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

    ```
    oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
    ```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - **openshift-storage.cephfs.csi.ceph.com-ctrlplugin-***
   - **openshift-storage.rbd.csi.ceph.com-ctrlplugin-***
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      oc debug node/<node_name>
      ```

      ```
      chroot /host
      ```

   b. Display the list of available block devices:, using the **lsblk** command.
      Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact IBM Support.

Replacing a failed Google Cloud Platform node on installer-provisioned infrastructure

Use this information to replace a failed GCP node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the faulty node that you need to replace and click on its Machine Name.
3. Go to Actions > Edit Annotations and click Add More.
4. Add **machine.openshift.io/exclude-node-draining**, and click Save.
5. Go to Action menu > Delete Machine and click Delete.
   A new machine is automatically created, wait for new machine to start.
   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
6. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
7. Apply the Fusion Data Foundation label to the new node using one of the following steps:

   From the user interface
   a. Go to Action Menu > Edit Labels > .
   b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

   From the command-line interface
   Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

   ```
   oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

8. Optional: If the failed Google Cloud instance is not removed automatically, terminate the instance from Google Cloud console.

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - **openshift-storage.cephfs.csi.ceph.com-ctrlplugin-***
   - **openshift-storage.rbd.csi.ceph.com-ctrlplugin-***
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   oc debug node/<node_name>
   ```

   ```
   chroot /host
   ```

   b. Display the list of available block devices:, using the **lsblk** command.
   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact IBM Support.

Replacing nodes on Fusion Data Foundation using local storage devices

Use this information to learn how to replace nodes on Fusion Data Foundation using local storage devices.

- See Fusion Data Foundation storage node failure to replace storage nodes on Fusion Data Foundation using local storage devices on the following infrastructures:
  - Bare metal
  - VMware
  - IBM Z or LinuxONE
  - IBM Power

Replacing devices

Explains how to safely replace storage devices for Fusion Data Foundation.

Replace any of the following storage devices, depending on the type of your deployment:

- AWS
- VMware
- Microsoft Azure
- Google Cloud
- Using local storage devices

- **Dynamically provisioned Fusion Data Foundation deployed on AWS**
  When replacing a device in a dynamically created storage cluster on an AWS installer-provisioned or user-provisioned infrastructure, you must replace the storage node. This can be done on either an operational or a failed AWS node.
- **Dynamically provisioned Fusion Data Foundation deployed on VMware**
  Replace an operational or failed storage device on VMware infrastructure.
- **Dynamically provisioned Fusion Data Foundation deployed on Azure**
  When replacing a device in a dynamically created storage cluster on an Azure installer-provisioned infrastructure, you must replace the storage node. This can be done on either an operational or a failed Azure node.
- **Dynamically provisioned Fusion Data Foundation deployed on Google Cloud Platform**
  When replacing a device in a dynamically created storage cluster on a Google Cloud Platform (GCP) installer-provisioned infrastructure, you must replace the storage node. You can perform this activity on either an operational or a failed GCP node.
- **Fusion Data Foundation deployed using local storage devices**
  Use this information to learn how to replace devices on Fusion Data Foundation using local storage devices.

Dynamically provisioned Fusion Data Foundation deployed on AWS

When replacing a device in a dynamically created storage cluster on an AWS installer-provisioned or user-provisioned infrastructure, you must replace the storage node. This can be done on either an operational or a failed AWS node.

- **Replacing an operational AWS node on user-provisioned infrastructure**
  Use this information to replace an operational AWS node on a user-provisioned infrastructure.
- **Replacing an operational AWS node on installer-provisioned infrastructure**
  Use this information to replace an operational AWS node on an installer-provisioned infrastructure.
- **Replacing a failed AWS node on user-provisioned infrastructure**
  Use this information to replace failed AWS node on a user-provisioned infrastructure.
- **Replacing a failed AWS node on installer-provisioned infrastructure**
  Use this information to replace a failed AWS node on an installer-provisioned infrastructure.

Replacing an operational AWS node on user-provisioned infrastructure

Use this information to replace an operational AWS node on a user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node that you need to replace.
2. Mark the node as unscheduable, where *<node_name>* specifies the name of node that you need to replace.

   ```
   oc adm cordon <node_name>
   ```

3. Drain the node.

   ```
   oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
   ```

   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
4. Delete the node.

   ```
   oc delete nodes <node_name>
   ```

5. Create a new Amazon Web Service (AWS) machine instance with the required infrastructure.
   For more information, see Platform requirements.
6. Create a new OpenShift Container Platform node using the new AWS machine instance.
7. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in a *Pending* state.

   ```
   oc get csr
   ```

8. Approve all the required OpenShift Container Platform CSRs for the new node, where *<certificate_name>* specifies the name of the CSR.

   ```
   oc adm certificate approve <certificate_name>
   ```

9. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
10. Apply the Fusion Data Foundation label to the new node using one of the following steps:

    From the user interface

a. Go to Action Menu > Edit Labels > .
b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

From the command-line interface

Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

```
oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      oc debug node/<node_name>
      ```

      ```
      chroot /host
      ```

   b. Display the list of available block devices:, using the **lsblk** command.
      Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Replacing an operational AWS node on installer-provisioned infrastructure

Use this information to replace an operational AWS node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the node that you need to replace.
   Take a note of its Machine Name.
3. Mark the node as unschedulable, using the following command, where *<node_name>* specifies the name of the node that you need to replace.

   ```
   oc adm cordon <node_name>
   ```

4. Drain the node.

   ```
   oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
   ```

   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
5. Click Compute > Machines and search for the required machine.
6. For the required machine, click Action menu > Delete Machine.
7. Click **Delete** to confirm that the machine is deleted.
   A new machine is automatically created.
8. Wait for the new machine to start and transition into the *Running* state.
   Important: This activity might take at least 5 - 10 minutes or more.
9. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
10. Apply the Fusion Data Foundation label to the new node using one of the following steps:

    From the user interface

    a. Go to Action Menu > Edit Labels > .
    b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

    From the command-line interface

    Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

    ```
    oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
    ```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:

- `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
- `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`

3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   oc debug node/<node_name>
   ```

   ```
   chroot /host
   ```

   b. Display the list of available block devices:, using the **lsblk** command.
   Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Replacing a failed AWS node on user-provisioned infrastructure

Use this information to replace failed AWS node on a user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the Amazon Web Service (AWS) machine instance of the node that you need to replace.
2. Log in to AWS, and terminate the AWS machine instance that you identified.
3. Create a new AWS machine instance with the required infrastructure.
   For more information, see Platform requirements.
4. Create a new OpenShift Container Platform node using the new AWS machine instance.
5. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in *Pending* state.

```
oc get csr
```

6. Approve all the required OpenShift Container Platform CSRs for the new node, where *<certificate_name>* specifies the name of the CSR.

```
oc adm certificate approve <certificate_name>
```

7. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
8. Apply the Fusion Data Foundation label to the new node using one of the following steps:

   From the user interface
   a. Go to Action Menu > Edit Labels > .
   b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

   From the command-line interface
   Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

   ```
   oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

```
oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   oc debug node/<node_name>
   ```

   ```
   chroot /host
   ```

   b. Display the list of available block devices:, using the **lsblk** command.

Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact IBM Support.

Replacing a failed AWS node on installer-provisioned infrastructure

Use this information to replace a failed AWS node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the faulty node that you need to replace and click on its Machine Name.
3. Go to Actions > Edit Annotations and click Add More.
4. Add **machine.openshift.io/exclude-node-draining**, and click Save.
5. Go to Action menu > Delete Machine and click Delete.
   A new machine is automatically created, wait for new machine to start.
   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
6. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
7. Apply the Fusion Data Foundation label to the new node using one of the following steps:

   From the user interface
   a. Go to Action Menu > Edit Labels > .
   b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

   From the command-line interface
   Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

   ```
   oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

8. Optional: If the failed Amazon Web Service (AWS) instance is not removed automatically, terminate the instance from the AWS console.

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - **openshift-storage.cephfs.csi.ceph.com-ctrlplugin-***
   - **openshift-storage.rbd.csi.ceph.com-ctrlplugin-***
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

   ```
   oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   oc debug node/<node_name>
   ```

   ```
   chroot /host
   ```

   b. Display the list of available block devices:, using the **lsblk** command.
   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact IBM Support.

Dynamically provisioned Fusion Data Foundation deployed on VMware

Replace an operational or failed storage device on VMware infrastructure.

- **Replacing operational or failed storage devices on VMware infrastructure**
  Use this information to replace an operational or failed storage devices on VMware infrastructure.

Replacing operational or failed storage devices on VMware infrastructure

Use this information to replace an operational or failed storage devices on VMware infrastructure.

Before you begin

- Ensure that the data is resilient.
- In the OpenShift Web Console, click Storage > Data Foundation.
- Click the Storage Systems tab, and then click **ocs-storagecluster-storagesystem**.
- In the Status card of Block and File dashboard, under the Overview tab, verify that *Data Resiliency* has a green tick mark.

About this task

Create a new Persistent Volume Claim (PVC) on a new volume, and remove the old object storage device (OSD) when one or more virtual machine disks (VMDK) needs to be replaced in Fusion Data Foundation which is deployed dynamically on VMware infrastructure.

Procedure

1. Identify the OSD that needs to be replaced and the OpenShift Container Platform node that has the OSD scheduled on it.

   ```
   oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
   ```

   Example output:

   ```
   rook-ceph-osd-0-6d77d6c7c6-m8xj6    0/1    CrashLoopBackOff    0    24h    10.129.0.16    compute-2    <none>
   <none>
   rook-ceph-osd-1-85d99fb95f-2svc7    1/1    Running             0    24h    10.128.2.24    compute-0    <none>
   <none>
   rook-ceph-osd-2-6c66cdb977-jp542    1/1    Running             0    24h    10.130.0.18    compute-1    <none>
   <none>
   ```

   In this example, **rook-ceph-osd-0-6d77d6c7c6-m8xj6** needs to be replaced and **compute-2** is the OpenShift Container platform node on which the OSD is scheduled.
   Note: If the OSD to be replaced is healthy, the status of the pod is **Running**.
2. Scale down the OSD deployment for the OSD to be replaced.
   Each time you want to replace the OSD, update the osd_id_to_remove parameter with the OSD ID, and repeat this step.

   ```
   $ osd_id_to_remove=0
   oc scale -n openshift-storage deployment rook-ceph-osd-${osd_id_to_remove} --replicas=0
   ```

   where, *osd_id_to_remove* is the integer in the pod name immediately after the **rook-ceph-osd** prefix. In this example, the deployment name is **rook-ceph-osd-0**.
   Example output:

   ```
   deployment.extensions/rook-ceph-osd-0 scaled
   ```

3. Verify that the **rook-ceph-osd** pod is terminated.

   ```
   oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
   ```

   Example output:

   ```
   No resources found.
   ```

   Important: If the **rook-ceph-osd** pod is in the *terminating* state, use the force option to delete the pod.

   ```
   oc delete pod rook-ceph-osd-0-6d77d6c7c6-m8xj6 --force --grace-period=0
   ```

   Example output:

   ```
   warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may con
     pod "rook-ceph-osd-0-6d77d6c7c6-m8xj6" force deleted
   ```

4. Remove the old OSD from the cluster so that you can add a new OSD.
   a. Delete any old **ocs-osd-removal** jobs.

      ```
      oc delete -n openshift-storage job ocs-osd-removal-job
      ```

      ```
      job.batch "ocs-osd-removal-job" deleted
      ```

   b. Go to the **openshift-storage** project.

      ```
      oc project openshift-storage
      ```

   c. Remove the old OSD from the cluster.

      ```
      oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=${osd_id_to_remove} -p FORCE_OSD_REMOVAL=false |oc
      create -n openshift-storage -f -
      ```

      The **FORCE_OSD_REMOVAL** value must be changed to **true** in clusters that only have three OSDs, or clusters with insufficient space to restore all three replicas of the data after the OSD is removed.

      Warning: This step results in OSD being completely removed from the cluster. Ensure that the correct value of **osd_id_to_remove** is provided.
5. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.
   A status of *Completed* confirms that the OSD removal job succeeded.

   ```
   oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
   ```

6. Ensure that the OSD removal is completed

   ```
   oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
   ```

   Example output:

   ```
   2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
   ```

Important: If the **ocs-osd-removal-job** pod fails and the pod is not in the expected *Completed* state, check the pod logs for further debugging.
For example:

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

7. If encryption was enabled at the time of install, remove **dm-crypt** managed **device-mapper** mapping from the OSD devices that are removed from the respective Fusion Data Foundation nodes.
   a. Get the PVC name(s) of the replaced OSD(s) from the logs of **ocs-osd-removal-job** pod.

   ```
   oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1  |egrep -i 'pvc|deviceset'

   2021-05-12 14:31:34.666000 I | cephosd: removing the OSD PVC "ocs-deviceset-xxxx-xxx-xxx-xxx"
   ```

   b. For each of the previously identified nodes, do the following:
      i. Create a **debug** pod and **chroot** to the host on the storage node.

      ```
      oc debug node/<node name>
      ```

      where *<node name>* is the name of the node.

      ```
      $ chroot /host
      ```

      ii. Find a relevant device name based on the PVC names identified in the previous step.

      ```
      dmsetup ls| grep <pvc name>
      ```

      where *<pvc name>* is the name of the PVC.

      Example output:

      ```
      ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt (253:0)
      ```

      iii. Remove the mapped device.

      ```
      $ cryptsetup luksClose --debug --verbose ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt
      ```

      Important: If the previous command gets stuck due to insufficient privileges, run the following commands:
      1. Press **CTRL+Z** to exit the previous command.
      2. Find the PID of the process which was stuck.

         ```
         $ ps -ef | grep crypt
         ```

      3. Terminate the process using the **kill** command.

         ```
         kill -9 <PID>
         ```

         where *<PID>*is the process ID.

      4. Verify that the device name is removed.

         ```
         $ dmsetup ls
         ```

8. Delete the **ocs-osd-removal** job.

   ```
   oc delete -n openshift-storage job ocs-osd-removal-job
   ```

   Example output:

   ```
   job.batch "ocs-osd-removal-job" deleted
   ```

   Note: When using an external key management system (KMS) with data encryption, the old OSD encryption key can be removed from the Vault server as it is now an orphan key.

What to do next

1. Verify that there is a new OSD running.

   ```
   oc get -n openshift-storage pods -l app=rook-ceph-osd
   ```

   Example output:

   ```
   rook-ceph-osd-0-5f7f4747d4-snshw                                  1/1     Running     0          4m47s
   rook-ceph-osd-1-85d99fb95f-2svc7                                  1/1     Running     0          1d20h
   rook-ceph-osd-2-6c66cdb977-jp542                                  1/1     Running     0          1d20h
   ```

2. Verify that there is a new PVC created which is in the *Bound* state.

   ```
   oc get -n openshift-storage pvc
   ```

   Example output:

   ```
   NAME                      STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
   ocs-deviceset-0-0-2s6w4   Bound    pvc-7c9bcaf7-de68-40e1-95f9-0b0d7c0ae2fc   512Gi      RWO            thin
   5m
   ocs-deviceset-1-0-q8fwh   Bound    pvc-9e7e00cb-6b33-402e-9dc5-b8df4fd9010f   512Gi      RWO            thin
   1d20h
   ocs-deviceset-2-0-9v8lq   Bound    pvc-38cdfcee-ea7e-42a5-a6e1-aaa6d4924291   512Gi      RWO            thin
   1d20h
   ```

3. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   a. Identify the nodes where the new OSD pods are running, where *<OSD-pod-name>* is the name of the OSD pod.

```
oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-pod-name>
```

For example:

```
oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

Example output:

```
NODE
compute-1
```

b. For each of the nodes identified in the previous step, do the following:

i. Create a debug pod and open a chroot environment for the selected host(s), where *<node name>* is the name of the node.

```
oc debug node/<node name>
```

```
$ chroot /host
```

ii. Check for the `crypt` keyword beside the `ocs-deviceset` name(s).

```
$ lsblk
```

4. Log in to OpenShift Web Console and view the storage dashboard.

Dynamically provisioned Fusion Data Foundation deployed on Azure

When replacing a device in a dynamically created storage cluster on an Azure installer-provisioned infrastructure, you must replace the storage node. This can be done on either an operational or a failed Azure node.

- **Replacing operational nodes on Azure installer-provisioned infrastructure**
  Use this information to replace an operational Azure node on an installer-provisioned infrastructure.
- **Replacing failed nodes on Azure installer-provisioned infrastructure**
  Use this information to replace a failed Azure node on an installer-provisioned infrastructure.

Replacing operational nodes on Azure installer-provisioned infrastructure

Use this information to replace an operational Azure node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the faulty node that you need to replace.
   Take note of its Machine Name.
3. Mark the node as unschedulable, where *<node_name>* specifies the name of node that you need to replace.

   ```
   oc adm cordon <node_name>
   ```

4. Drain the node.

   ```
   oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
   ```

   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
5. Click Compute > Machines and search for the required machine.
6. For the required machine, click Action menu > Delete Machine.
7. Click **Delete** to confirm that the machine is deleted.
   A new machine is automatically created.
8. Wait for the new machine to start and transition into *Running* state.
   Important: This activity might take at least 5 - 10 minutes or more.
9. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
10. Apply the Fusion Data Foundation label to the new node using one of the following steps:

    From the user interface
    a. Go to Action Menu > Edit Labels > .
    b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

    From the command-line interface
    Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

    ```
    oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
    ```

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

   ```
   oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`

- `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`

3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   oc debug node/<node_name>
   ```

   ```
   chroot /host
   ```

   b. Display the list of available block devices:, using the **lsblk** command.
   Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Replacing failed nodes on Azure installer-provisioned infrastructure

Use this information to replace a failed Azure node on an installer-provisioned infrastructure.

Procedure

1. Log in to the OpenShift Web Console, and click Compute > Nodes.
2. Identify the faulty node that you need to replace and click on its Machine Name.
3. Go to Actions > Edit Annotations and click Add More.
4. Add `machine.openshift.io/exclude-node-draining`, and click Save.
5. Go to Action menu > Delete Machine and click Delete.
   A new machine is automatically created, wait for new machine to start.
   Important: This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.
6. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
7. Apply the Fusion Data Foundation label to the new node using one of the following steps:

   From the user interface
   a. Go to Action Menu > Edit Labels > .
   b. Add `cluster.ocs.openshift.io/openshift-storage`, and click **Save**.

   From the command-line interface
   Apply the Fusion Data Foundation label to the new node:, where *<new_node_name>* specifies the name of the new node.

   ```
   oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

8. Optional: If the failed Azure instance is not removed automatically, terminate the instance from the Azure console.

What to do next

Verify that the new node and all pods are running.

1. Verify that the new node is present in the output:

```
oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
   - `openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*`
   - `openshift-storage.rbd.csi.ceph.com-ctrlplugin-*`
3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   oc debug node/<node_name>
   ```

   ```
   chroot /host
   ```

   b. Display the list of available block devices:, using the **lsblk** command.
   Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

6. If the verification steps fail, contact IBM Support.

Dynamically provisioned Fusion Data Foundation deployed on Google Cloud Platform

When replacing a device in a dynamically created storage cluster on a Google Cloud Platform (GCP) installer-provisioned infrastructure, you must replace the storage node. You can perform this activity on either an operational or a failed GCP node.

- Replacing an operational node on Google Cloud Platform installer-provisioned infrastructure
- Replacing a failed Google Cloud Platform node on installer-provisioned infrastructure

Fusion Data Foundation deployed using local storage devices

Use this information to learn how to replace devices on Fusion Data Foundation using local storage devices.

- See Disk replacement and OpenShift dedicated OSD failure recovery to replace storage devices on Fusion Data Foundation using local storage devices on the following infrastructures:
  - Bare metal
  - VMware
  - IBM Z or LinuxONE
  - IBM Power

Monitoring Fusion Data Foundation

Learn how to monitor IBM Fusion Data Foundation using the Block and File, and Object dashboards.

- **Cluster health**
  Learn how to monitor the cluster health of IBM Fusion Data Foundation and storage.
- **Multicluster storage health**
  To view the overall storage health status across all the clusters with Fusion Data Foundation and manage its capacity, you must first enable the multicluster dashboard on the Hub cluster.
- **Metrics**
  Learn how to navigate and understand the dashboard metrics in the in the OpenShift Web Console.
- **Alerts**
  Internal mode cluster alerts are displayed in the dashboard and the OpenShift Container Platform.
- **Fusion Data Foundation metrics**
  Reference guide for ceph-exporter, rook, and NooBaa metrics to improve observability and simplify metric interpretation in Fusion Data Foundation.

Cluster health

Learn how to monitor the cluster health of IBM Fusion Data Foundation and storage.

- **Verifying Fusion Data Foundation is healthy**
  Storage health is visible on the Block and File and Object dashboards.
- **Storage health levels and cluster state**
  Status information and alerts related to Fusion Data Foundation are displayed in the storage dashboards.
- **Viewing Fusion Data Foundation infrastructure health**
  Monitor the overall health of your infrastructure and manage health checks to ensure optimal system performance.

Verifying Fusion Data Foundation is healthy

Storage health is visible on the Block and File and Object dashboards.

Procedure

1. In the OpenShift Web Console, click Storage > Data Foundation.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
3. Check if the **Status** card has a green tick in the **Block and File** and the **Object** tabs.
   Green tick indicates that the cluster is healthy.

What to do next

See Storage health levels and cluster state for information about the different health states and the alerts that appear.

Storage health levels and cluster state

Status information and alerts related to Fusion Data Foundation are displayed in the storage dashboards.

Block and File dashboard indicators

The Block and File dashboard shows the complete state of Fusion Data Foundation and the state of persistent volumes.

Table 1 describes states that are possible for each resource type.

Table 1. Fusion Data Foundation health levels

State	Icon	Description
UNKNOWN	Question mark	Fusion Data Foundation is not deployed or unavailable.
Healthy	Green tick	Cluster health is good.
Warning	Yellow exclamation in a triangle	Fusion Data Foundation cluster is in a warning state. In internal mode, an alert is displayed along with the issue details. Alerts are not displayed for external mode.
Error	Red exclamation in a circle	Fusion Data Foundation cluster has encountered an error and some component is nonfunctional. In internal mode, an alert is displayed along with the issue details. Alerts are not displayed for external mode.

Object dashboard indicators

The Object dashboard shows the state of the Multicloud Object Gateway and any object claims in the cluster.

Table 2 describes states that are possible for each resource type.

Table 2. Object Service health levels

State	Description
Green Tick	Object storage is healthy.
`Multicloud Object Gateway is not running`	Shown when NooBaa system is not found.
`All resources are unhealthy`	Shown when all NooBaa pools are unhealthy.
`Many buckets have issues`	Shown when >= 50% of buckets encounter error(s).
`Some buckets have issues`	Shown when >= 30% of buckets encounter error(s).
`Unavailable`	Shown when network issues and/or errors exist.

Alert panel

The Alert panel appears below the **Status** card in both the **Block and File** dashboard and the **Object** dashboard when the cluster state is not healthy.

For more information about alerts and how to respond to them , see Troubleshooting alerts and errors in Fusion Data Foundation.

Viewing Fusion Data Foundation infrastructure health

Monitor the overall health of your infrastructure and manage health checks to ensure optimal system performance.

About this task

This section details how to monitor the overall health of your infrastructure and temporarily disable health checks when needed.

Procedure

1. In the OpenShift Web Console, click Storage > Data Foundation.
2. In the Overview tab, view the Infrastructure health card.
   Here you see your infrastructure health score out of 100%. This score reflects the overall operational status of your Fusion Data Foundation cluster. Your system starts with a health score of 100%. Points are deducted whenever issues are found. The amount deducted depends on the severity of the issue:

   - Minor: subtract 2 points
   - Medium: subtract 10 points
   - Critical: subtract 20 points

   View health checks and alerts

   a. In the Infrastructure health card, click View health checks. Here you see any of the following alerts:

   Table 1. Infrastructure health checks

Health check	Severity	Description	Metrics	Notes / Requirements
Node Latency	Medium	Maximum latency between nodes running OSDs should be 10ms roundtrip (last 24h). Maximum latency between other ODF nodes should be 100ms roundtrip (last 24h).	`probe_icmp_duration_seconds`	Requires Prometheus Blackbox Exporter to be deployed and configured.
MTU Check	Minor	Verify that network interfaces used by ODF nodes have an MTU 9000 bytes.	`node_network_mtu_bytes`	-
Disk Busy	Medium	Checks if disks used by ODF are > 90% busy on average.	`node_disk_io_time_seconds_total`	-
ODF Pod Restarts	Medium	Detect if any core ODF pods have restarted within the last 24 hours.	`kube_pod_container_status_restarts_total`	-

Health check	Severity	Description	Metrics	Notes / Requirements
NIC Bandwidth Saturation	Medium	Detect if any ODF node NIC used for cluster traffic exceeded 90% bandwidth utilization over a 5 minute period within the last 24 hours.	• `node_network_receive_bytes_total` • `node_network_transmit_bytes_total` • `node_network_speed_bytes`	-

Table 2. Infrastructure health alerts

Alert	Severity
ODFNodeLatencyHighOnOSDNodes	Medium
ODFNodeLatencyHighOnNonOSDNodes	Medium
ODFNodeMTULessThan9000	Minor
ODFDiskUtilizationHigh	Medium
ODFCorePodRestarted	Medium
ODFNodeNICBandwidthSaturation	Medium

[Optional] Silence health checks

a. In the Infrastructure health card, click View health checks.
b. View the list of health checks and select the checkbox next to the health checks you want to silence.
c. Click Silence at the end of the row.
d. Enter a Duration and click Silence.

Multicluster storage health

To view the overall storage health status across all the clusters with Fusion Data Foundation and manage its capacity, you must first enable the multicluster dashboard on the Hub cluster.

- **Enabling multicluster dashboard on Hub cluster**
  You can enable the multicluster dashboard on the install screen either before or after installing ODF Multicluster Orchestrator with the console plugin.
- **Verifying multicluster storage health on hub cluster**
  Follow this procedure to verify multicluster storage health on hub cluster.

Enabling multicluster dashboard on Hub cluster

You can enable the multicluster dashboard on the install screen either before or after installing ODF Multicluster Orchestrator with the console plugin.

Before you begin

- Ensure that you have installed OpenShift Container Platform version 4.16 or higher version and have administrator privileges.
- Ensure that you have installed Red Hat Advanced Cluster Management (RHACM) for Kubernetes 2.9 or higher version from Operator Hub. For instructions on how to install, see Installation and upgrade > Install in Red Hat Advanced Cluster Management for Kubernetes product documentation.
- Ensure you have enabled observability on RHACM. See Operations > Observability > Observability service > Enabling the Observability service > Enabling Observability from the command line interface in Red Hat Advanced Cluster Management for Kubernetes product documentation.

Procedure

1. Create the configmap file named observability-metrics-custom-allowlist.yaml and add the name of the custom metric to the metrics_list.yaml parameter.
   You can use the following YAML to list the Fusion Data Foundation metrics on Hub cluster. For details, see Operations > Observability > Observability service > Observability advanced confihuration > Adding custom metrics in Red Hat Advanced Cluster Management for Kubernetes product documentation.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: observability-metrics-custom-allowlist
  Namespace: open-cluster-management-observability
data:
  metrics_list.yaml: |
    names:
      - odf_system_health_status
      - odf_system_map
      - odf_system_raw_capacity_total_bytes
      - odf_system_raw_capacity_used_bytes
    matches:
      - __name__="csv_succeeded",exported_namespace="openshift-storage",name=~"odf-operator.*"
```

2. Run the following command in the **open-cluster-management-observability** namespace:

```
oc apply -n open-cluster-management-observability -f observability-metrics-custom-allowlist.yaml
```

   After the observability-metrics-custom-allowlist yaml is created, RHACM starts collecting the listed Fusion Data Foundation metrics from all the managed clusters.

If you want to exclude specific managed clusters from collecting the observability data, add the following cluster label to your clusters: observability: disabled.

3. To view the multicluster health, see Verifying Fusion Data Foundation is healthy.

Verifying multicluster storage health on hub cluster

Follow this procedure to verify multicluster storage health on hub cluster.

Before you begin

Ensure that you have enabled multicluster monitoring, as described in Enabling multicluster dashboard on Hub cluster.

Procedure

1. In the OpenShift web console of Hub cluster, ensure All Clusters is selected.
2. Go to Data Services > Storage System.
3. On the Overview tab, verify that there are green ticks in front of Fusion Data Foundation and Systems. This indicates that the operator is running and all storage systems are available.
4. From the Status card, view the operator and storage system statuses.
   - View the operator status by clicking Fusion Data Foundation.
   - View the storage system status by clicking Systems.

   The **Storage system capacity** card shows the following details:
   - Name of the storage system
   - Cluster name
   - Graphical representation of total and used capacity in percentage
   - Actual values for total and used capacity in TiB

Metrics

Learn how to navigate and understand the dashboard metrics in the in the OpenShift Web Console.

- **Metrics in the Block and File dashboard**
  Learn how to navigate Block and File dashboard in the OpenShift Web Console.
- **Metrics in the Object dashboard**
  Learn how to navigate the Object dashboard.
- **Pool metrics**
  The Pool metrics dashboard provides information to ensure efficient data consumption, and how to enable or disable compression if less effective.
- **Network File System metrics**
  View NFS throughput, client connections, and Ganesha server health using the Network File System metrics dashboard in the OpenShift Web Console.
- **Enabling metadata on RBD and CephFS volumes**
  Enable PVC, PV, and namespace metadata on RBD and CephFS volumes by setting the `CSI_ENABLE_METADATA` variable in the `rook-ceph-operator-config` ConfigMap.

Metrics in the Block and File dashboard

Learn how to navigate Block and File dashboard in the OpenShift Web Console.

You can navigate to the Block and File dashboard in the OpenShift Web Console as follows:

1. Go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link.
3. Click the Block and File tab.

The following cards on the Block and File dashboard provides the metrics based on deployment mode (internal or external):

Details card
: The Details card shows the following:

  - Service Name
  - Cluster name
  - The name of the Provider on which the system runs (example: `AWS`, `VSphere`, `None` for Bare metal)
  - Mode (deployment mode as either Internal or External)
  - Fusion Data Foundation operator version.
  - In-transit encryption (shows whether the encryption is enabled or disabled)

Inventory card
: The Inventory card shows the number of active nodes, PVCs and PVs backed by Fusion Data Foundation provisioner. On the left hand side of the card, total number of storage nodes, PVCs and PVs are displayed. While on the corresponding right hand side of the card, number of storage nodes in *Not Ready* state, count of PVCs in *Pending* state and PVs in Released state are shown.
  Note: For external mode, the number of nodes is `0` by default, since there are no dedicated nodes for Fusion Data Foundation.

Status card
: This card shows whether the cluster is up and running without any errors or is experiencing some issues.

For internal mode, Data Resiliency indicates the status of data re-balancing in Ceph across the replicas. When the internal mode cluster is in a warning or error state, the Alerts section is shown along with the relevant alerts.

Note: For external mode, Data Resiliency and alerts are not displayed.

Raw Capacity card

This card shows the total raw storage capacity which includes replication, on the cluster.

Note: This card is not applicable for external mode clusters.

- Used legend indicates space used raw storage capacity on the cluster
- Available legend indicates the available raw storage capacity on the cluster.

Consumption trend card

This card shows the storage consumption rate in GiB per day, which is calculated based on the actual capacity utilization, historical usage, and current consumption rate. Also, the card shows the number of days left for the storage to reach the threshold capacity.

Used Capacity Breakdown card

This card shows the actual amount of non-replicated data stored in the cluster and its distribution. You can choose between Projects, Storage Classes, and Pods from the drop-down menu on the card. These options are for filtering the data shown in the graph. The graph displays the used capacity for only the top five entities, based on usage. The aggregate usage of the remaining entities is displayed as `Other`.

Option	Display
Projects	The aggregated capacity of each project which is using the Fusion Data Foundation and how much is being used.
Storage Classes	The aggregate capacity based on Fusion Data Foundation based storage classes.
Pods	All the pods trying to use the PVC backed by Fusion Data Foundation provisioner.

Note: For external mode, see Capacity breakdown card.

Capacity breakdown card

This card is only applicable for external mode clusters. In this card, you can view graphic breakdown of capacity per project, storage classes and pods. You can choose between Projects, Storage Classes, and Pods from the drop-down menu on the card. These options are for filtering the data shown in the graph. The graph displays the used capacity for only the top five entities, based on usage. The aggregate usage of the remaining entities is displayed as `Other`.

Utilization card

The card shows used capacity, input/output operations per second, latency, throughput, and recovery information for the internal mode cluster.

For external mode, this card shows only the used and requested capacity details for that cluster.

Storage Efficiency card

This card shows the compression ratio that represents a compressible data effectiveness metric inclusive of all compression-enabled pools. It also shows the savings metric that represents the actual disk capacity saved inclusive of all compression-enabled pools and associated replicas.

Activity card

This card shows what activities are happening or have recently happened in the Fusion Data Foundation cluster. The card is separated into two sections:

Ongoing

Displays the progress of ongoing activities related to rebuilding of data resiliency and upgrading of Fusion Data Foundation operator.

Recent Events

Displays the list of events that happened in the **`openshift-storage`** namespace.

Metrics in the Object dashboard

Learn how to navigate the Object dashboard.

You can navigate to the Object dashboard in the OpenShift Web Console as follows:

1. Go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link.
3. Click the Object tab.

The following metrics are available in the Object dashboard:

Details card

This card shows the following information:

Service Name

The Multicloud Object Gateway (MCG) service name.

System Name

The Multicloud Object Gateway and RADOS Object Gateway system names. The Multicloud Object Gateway system name is also a hyperlink to the MCG management user interface.

Provider

The name of the provider on which the system runs (for example: **`AWS`**, **`VSphere`**, **`None`** for Baremetal)

Version

Fusion Data Foundation operator version.

Storage Efficiency card

In this card you can view how the MCG optimizes the consumption of the storage backend resources through deduplication and compression and provides you with a calculated efficiency ratio (application data vs logical data) and an estimated savings figure (how many bytes the MCG did not send to the storage provider) based on capacity of bare metal and cloud based storage and egress of cloud based storage.

Buckets card

Buckets are containers maintained by the MCG and RADOS Object Gateway to store data on behalf of the applications. These buckets are created and accessed through object bucket claims (OBCs). A specific policy can be applied to bucket to customize data placement, data spill-over, data resiliency, capacity quotas, and so on.

In this card, information about object buckets (OB) and object bucket claims (OBCs) is shown separately. OB includes all the buckets that are created using S3 or the user interface(UI) and OBC includes all the buckets created using YAMLs or the command line interface (CLI). The number displayed on the left of the bucket type is the total count of OBs or OBCs. The number displayed on the right shows the error count and is visible only when the error count is greater than zero. You can click on the number to see the list of buckets that has the warning or error status.

Resource Providers card
: This card displays a list of all Multicloud Object Gateway and RADOS Object Gateway resources that are currently in use. Those resources are used to store data according to the buckets policies and can be a cloud-based resource or a bare metal resource.

Status card
: This card shows whether the system and its services are running without any issues. When the system is in a warning or error state, the alerts section is shown and the relevant alerts are displayed there. Click the alert links beside each alert for more information about the issue. For information about health checks, see Cluster health.

  If multiple object storage services are available in the cluster, click the service type (such as Object Service or Data Resiliency) to see the state of the individual services.

  Data resiliency in the status card indicates if there is any resiliency issue regarding the data stored through the Multicloud Object Gateway and RADOS Object Gateway.

Capacity breakdown card
: In this card you can visualize how applications consume the object storage through the Multicloud Object Gateway and RADOS Object Gateway. You can use the Service Type drop-down to view the capacity breakdown for the Multicloud Gateway and Object Gateway separately. When viewing the Multicloud Object Gateway, you can use the Break By drop-down to filter the results in the graph by either Projects or Bucket Class.

Performance card
: In this card, you can view the performance of the Multicloud Object Gateway or RADOS Object Gateway. Use the Service Type drop-down to choose which you would like to view.

  For Multicloud Object Gateway accounts, you can view the I/O operations and logical used capacity. For providers, you can view I/O operation, physical and logical usage, and egress.

  Table 1 explains the different metrics that you can view based on your selection from the drop-down menus on the card.

Table 1. Indicators for Multicloud Object Gateway

Consumer types	Metrics	Chart display
Accounts	I/O operations	Displays read and write I/O operations for the top five consumers. The total reads and writes of all the consumers is displayed at the bottom. This information helps you monitor the throughput demand (IOPS) per application or account.
Accounts	Logical Used Capacity	Displays total logical usage of each account for the top five consumers. This helps you monitor the throughput demand per application or account.
Providers	I/O operations	Displays the count of I/O operations generated by the MCG when accessing the storage backend hosted by the provider. This helps you understand the traffic in the cloud so that you can improve resource allocation according to the I/O pattern, thereby optimizing the cost.
Providers	Physical vs Logical usage	Displays the data consumption in the system by comparing the physical usage with the logical usage per provider. This helps you control the storage resources and devise a placement strategy in line with your usage characteristics and your performance requirements while potentially optimizing your costs.
Providers	Egress	The amount of data the MCG retrieves from each provider (read bandwidth originated with the applications). This helps you understand the traffic in the cloud to improve resource allocation according to the egress pattern, thereby optimizing the cost.

  For the RADOS Object Gateway, you can use the Metric drop-down to view the Latency or Bandwidth.

  Latency
  : Provides a visual indication of the average GET/PUT latency imbalance across RADOS Object Gateway instances.

  Bandwidth
  : Provides a visual indication of the sum of GET/PUT bandwidth across RADOS Object Gateway instances.

Activity card
: This card displays what activities are happening or have recently happened in the Fusion Data Foundation cluster. The card is separated into two sections:

  Ongoing
  : Displays the progress of ongoing activities related to rebuilding of data resiliency and upgrading of Fusion Data Foundation operator.

  Recent Events
  : Displays the list of events that happened in the `openshift-storage` namespace.

Pool metrics

The Pool metrics dashboard provides information to ensure efficient data consumption, and how to enable or disable compression if less effective.

To view the pool list, from the OpenShift Web Console:

1. Go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System, select the storage system and then click **BlockPools**.

When you click on a pool name, the following cards on each Pool dashboard is displayed along with the metrics based on deployment mode (internal or external):

Details card
: The Details card shows the following:

  - Pool Name

- Volume type
- Replicas

Status card
: This card shows whether the pool is up and running without any errors or is experiencing some issues.

Mirroring card
: When the mirroring option is enabled, this card shows the mirroring status, image health, and last checked time-stamp. The mirroring metrics are displayed when cluster level mirroring is enabled. The metrics help to prevent disaster recovery failures and notify of any discrepancies so that the data is kept intact.

  The mirroring card shows high-level information such as:

  - Mirroring state as either enabled or disabled for the particular pool.
  - Status of all images under the pool as replicating successfully or not.
  - Percentage of images that are replicating and not replicating.

Inventory card
: The Inventory card shows the number of storage classes and Persistent Volume Claims.

Compression card
: This card shows the compression status as enabled or disabled as the case may be. It also displays the storage efficiency details as follows:

  - Compression eligibility that indicates what portion of written compression-eligible data is compressible (per ceph parameters)
  - Compression ratio of compression-eligible data
  - Compression savings provides the total savings (including replicas) of compression-eligible data

    For information on how to enable or disable compression for an existing pool, see Updating an existing pool.

Raw Capacity card
: This card shows the total raw storage capacity which includes replication, on the cluster.

  - Used legend indicates storage capacity used by the pool
  - Available legend indicates the available raw storage capacity on the cluster

Performance card
: In this card, you can view the usage of I/O operations and throughput demand per application or account. The graph indicates the average latency or bandwidth across the instances.

Network File System metrics

View NFS throughput, client connections, and Ganesha server health using the Network File System metrics dashboard in the OpenShift Web Console.

Before you begin

- OpenShift Container Platform is installed and you have administrative access to OpenShift Web Console.
- Ensure that NFS is enabled.

About this task

The Network File System (NFS) metrics dashboard provides enhanced observability for NFS mounts such as the following:

- Mount point for any exported NFS shares
- Number of client mounts
- A breakdown statistics of the clients that are connected to help determine internal versus the external client mounts
- Grace period status of the Ganesha server
- Health statuses of the Ganesha server

Procedure

1. Click Storage > Data Foundation.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
3. Click the **Network file system** tab. This tab is available only when NFS is enabled.
   Note: When you enable or disable NFS from command-line interface, you must perform hard refresh to display or hide the **Network file system** tab in the dashboard.
   The following NFS metrics are displayed:

   Status Card

   This card shows the status of the server based on the total number of active worker threads. Non-zero threads specify healthy status.

   Throughput Card

   This card shows the throughput of the server which is the summation of the total request bytes and total response bytes for both read and write operations of the server.

   Top client Card

This card shows the throughput of clients which is the summation of the total of the response bytes sent by a client and the total request bytes by a client for both read and write operations. It shows the top three of such clients.

Enabling metadata on RBD and CephFS volumes

Enable PVC, PV, and namespace metadata on RBD and CephFS volumes by setting the **CSI_ENABLE_METADATA** variable in the **rook-ceph-operator-config** ConfigMap.

Before you begin

- Ensure to install **ocs_operator** and create a **storagecluster** for the operator.
- Ensure that the **storagecluster** is in **Ready** state.

```
oc get storagecluster
```

```
NAME                 AGE   PHASE   EXTERNAL   CREATED AT             VERSION
ocs-storagecluster   57m   Ready              2022-08-30T06:52:58Z   4.12.0
```

About this task

You can set the persistent volume claim (PVC), persistent volume (PV), and Namespace names in the RADOS block device (RBD) and CephFS volumes for monitoring purposes. This enables you to read the RBD and CephFS metadata to identify the mapping between the OpenShift Container Platform and RBD and CephFS volumes.

To enable RADOS block device (RBD) and CephFS volume metadata feature, you need to set the **CSI_ENABLE_METADATA** variable in the **rook-ceph-operator-configconfigmap**. By default, this feature is disabled. If you enable the feature after upgrading from a previous version, the existing PVCs do not contain the metadata. Also, when you enable the metadata feature, the PVCs that were created before enabling do not have the metadata.

Procedure

1. Edit the **rook-ceph** operator **ConfigMap** to mark **CSI_ENABLE_METADATA** to **true**.

```
oc patch cm rook-ceph-operator-config -n openshift-storage -p $'data:\n "CSI_ENABLE_METADATA":  "true"'
configmap/rook-ceph-operator-config patched
```

2. Wait for the respective **openshift-storage.cephfs.csi.ceph.com-ctrlplugin-***, **openshift-storage.cephfs.csi.ceph.com-nodeplugin-***, **openshift-storage.rbd.csi.ceph.com-ctrlplugin-***, and **openshift-storage.rbd.csi.ceph.com-nodeplugin-*** pods to reach the **Running** state.
Note: Ensure that the **setmetadata** variable is automatically set after the metadata feature is enabled. This variable should not be available when the metadata feature is disabled.

```
oc get pods | grep csi
```

```
ceph-csi-controller-manager-667b84dcc4-9kpgb                      1/1     Running     0          5h5m
csi-addons-controller-manager-55cb46d5bd-fp4h5                    1/1     Running     0          5h5m
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-b6d89f579lwn4j   8/8     Running     0          4h59m
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-b6d89f579rjw9p   8/8     Running     0          4h59m
openshift-storage.cephfs.csi.ceph.com-nodeplugin-4dlmj            3/3     Running     0          4h59m
openshift-storage.cephfs.csi.ceph.com-nodeplugin-ffr4p            3/3     Running     0          4h59m
openshift-storage.cephfs.csi.ceph.com-nodeplugin-t6ccs            3/3     Running     0          4h59m
openshift-storage.rbd.csi.ceph.com-ctrlplugin-6487df698c-8h7bw    8/8     Running     0          4h59m
openshift-storage.rbd.csi.ceph.com-ctrlplugin-6487df698c-zp8k2    8/8     Running     0          4h59m
openshift-storage.rbd.csi.ceph.com-nodeplugin-csi-addons-bllss    2/2     Running     0          4h59m
openshift-storage.rbd.csi.ceph.com-nodeplugin-csi-addons-j6t8p    2/2     Running     0          4h59m
openshift-storage.rbd.csi.ceph.com-nodeplugin-csi-addons-klvrj    2/2     Running     0          4h59m
openshift-storage.rbd.csi.ceph.com-nodeplugin-kx89w               3/3     Running     0          4h59m
openshift-storage.rbd.csi.ceph.com-nodeplugin-mvlj2               3/3     Running     0          4h59m
openshift-storage.rbd.csi.ceph.com-nodeplugin-r9542               3/3     Running     0          4h59m
```

- **Verify the metadata for RBD PVC**
  Verify that metadata is correctly set on an RBD PVC by checking the image metadata in the Ceph CLI after creating a persistent volume claim.
- **Verify the metadata for RBD clone**
  Verify that metadata is correctly set on an RBD clone by checking the image metadata in the Ceph CLI after creating a PVC clone.
- **Verify the metadata for RBD snapshots**
  Verify that metadata is correctly set on an RBD snapshot by checking the image metadata in the Ceph CLI after creating a volume snapshot.
- **Verify the metadata for RBD Restore**
  Verify that metadata is correctly set on a restored RBD PVC by checking the image metadata in the Ceph CLI after restoring a volume snapshot.
- **Verify the metadata for CephFS PVC**
  Verify that metadata is correctly set on a CephFS PVC by checking the subvolume metadata in the Ceph CLI after creating a persistent volume claim.
- **Verify the metadata for CephFS clone**
  Verify that metadata is correctly set on a CephFS clone by checking the subvolume metadata in the Ceph CLI after creating a PVC clone.
- **Verify the metadata for CephFS volume snapshot**
  Verify that metadata is correctly set on a CephFS volume snapshot by checking the subvolume snapshot metadata in the Ceph CLI after creating a volume snapshot.
- **Verify the metadata of CephFS Restore**
  Verify that metadata is correctly set on a restored CephFS PVC by checking the subvolume metadata in the Ceph CLI after restoring a volume snapshot.

Verify the metadata for RBD PVC

Verify that metadata is correctly set on an RBD PVC by checking the image metadata in the Ceph CLI after creating a persistent volume claim.

Procedure

1. Create a PVC.

```
cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rbd-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ocs-storagecluster-ceph-rbd
EOF
```

2. Check the status of the PVC.

```
oc get pvc | grep cephfs
```

```
rbd-pvc                           Bound    pvc-30628fa8-2966-499c-832d-a6a3a8ebc594   1Gi        RWO            ocs-
storagecluster-ceph-rbd   32s
```

3. Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
rbd ls ocs-storagecluster-cephblockpool
```

```
csi-vol-7d67bfad-2842-11ed-94bd-0a580a830012
csi-vol-ed5ce27b-2842-11ed-94bd-0a580a830012
```

```
rbd image-meta ls ocs-storagecluster-cephblockpool/csi-vol-ed5ce27b-2842-11ed-94bd-0a580a830012
```

There are four metadata on this image:

```
Key                               Value
csi.ceph.com/cluster/name         6cd7a18d-7363-4830-ad5c-f7b96927f026
csi.storage.k8s.io/pv/name        pvc-30628fa8-2966-499c-832d-a6a3a8ebc594
csi.storage.k8s.io/pvc/name       rbd-pvc
csi.storage.k8s.io/pvc/namespace  openshift-storage
```

Verify the metadata for RBD clone

Verify that metadata is correctly set on an RBD clone by checking the image metadata in the Ceph CLI after creating a PVC clone.

Procedure

1. Create a clone.

```
cat <<EOF | oc create -f -apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rbd-pvc-clone
spec:
  storageClassName: ocs-storagecluster-ceph-rbd
  dataSource:
    name: rbd-pvc
    kind: PersistentVolumeClaim
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
EOF
```

2. Check the status of the clone.

```
oc get pvc | grep rbd-pvc
```

```
rbd-pvc                           Bound    pvc-30628fa8-2966-499c-832d-a6a3a8ebc594   1Gi        RWO            ocs-
storagecluster-ceph-rbd   15m
rbd-pvc-clone                     Bound    pvc-0d72afda-f433-4d46-a7f1-a5fcb3d766e0   1Gi        RWO            ocs-
storagecluster-ceph-rbd   52s
```

3. Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
rbd ls ocs-storagecluster-cephblockpool
```

```
csi-vol-063b982d-2845-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012-temp
csi-vol-7d67bfad-2842-11ed-94bd-0a580a830012
csi-vol-ed5ce27b-2842-11ed-94bd-0a580a830012
```

```
rbd image-meta ls ocs-storagecluster-cephblockpool/csi-vol-063b982d-2845-11ed-94bd-0a580a830012

There are 4 metadata on this image:

Key                                 Value
csi.ceph.com/cluster/name           6cd7a18d-7363-4830-ad5c-f7b96927f026
csi.storage.k8s.io/pv/name          pvc-0d72afda-f433-4d46-a7f1-a5fcb3d766e0
csi.storage.k8s.io/pvc/name         rbd-pvc-clone
csi.storage.k8s.io/pvc/namespace    openshift-storage
```

Verify the metadata for RBD snapshots

Verify that metadata is correctly set on an RBD snapshot by checking the image metadata in the Ceph CLI after creating a volume snapshot.

Procedure

1. Create a snapshot.

```
cat <<EOF | oc create -f -
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: rbd-pvc-snapshot
spec:
  volumeSnapshotClassName: ocs-storagecluster-rbdplugin-snapclass
  source:
    persistentVolumeClaimName: rbd-pvc
EOF
volumesnapshot.snapshot.storage.k8s.io/rbd-pvc-snapshot created
```

2. Check the status of the snapshot.

```
oc get volumesnapshot


NAME               READYTOUSE   SOURCEPVC   SOURCESNAPSHOTCONTENT   RESTORESIZE   SNAPSHOTCLASS
SNAPSHOTCONTENT                                    CREATIONTIME   AGE
rbd-pvc-snapshot   true         rbd-pvc                             1Gi           ocs-storagecluster-rbdplugin-snapclass
snapcontent-b992b782-7174-4101-8fe3-e6e478eb2c8f   17s            18s
```

3. Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
rbd ls ocs-storagecluster-cephblockpool

csi-snap-a1e24408-2848-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012-temp
csi-vol-7d67bfad-2842-11ed-94bd-0a580a830012
csi-vol-ed5ce27b-2842-11ed-94bd-0a580a830012

rbd image-meta ls ocs-storagecluster-cephblockpool/csi-snap-a1e24408-2848-11ed-94bd-0a580a830012


There are 4 metadata on this image:

Key                                        Value
csi.ceph.com/cluster/name                  6cd7a18d-7363-4830-ad5c-f7b96927f026
csi.storage.k8s.io/volumesnapshot/name     rbd-pvc-snapshot
csi.storage.k8s.io/volumesnapshot/namespace    openshift-storage
csi.storage.k8s.io/volumesnapshotcontent/name  snapcontent-b992b782-7174-4101-8fe3-e6e478eb2c8f
```

Verify the metadata for RBD Restore

Verify that metadata is correctly set on a restored RBD PVC by checking the image metadata in the Ceph CLI after restoring a volume snapshot.

Procedure

1. Restore a volume snapshot.

```
cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rbd-pvc-restore
spec:
  storageClassName: ocs-storagecluster-ceph-rbd
  dataSource:
    name: rbd-pvc-snapshot
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

```
EOF
persistentvolumeclaim/rbd-pvc-restore created
```

2. Check the status of the restored volume snapshot.

```
oc get pvc | grep rbd
```

```
db-noobaa-db-pg-0                  Bound    pvc-615e2027-78cd-4ea2-a341-fdedd50c5208   50Gi       RWO            ocs-
storagecluster-ceph-rbd   51m
rbd-pvc                            Bound    pvc-30628fa8-2966-499c-832d-a6a3a8ebc594   1Gi        RWO            ocs-
storagecluster-ceph-rbd   47m
rbd-pvc-clone                      Bound    pvc-0d72afda-f433-4d46-a7f1-a5fcb3d766e0   1Gi        RWO            ocs-
storagecluster-ceph-rbd   32m
rbd-pvc-restore                    Bound    pvc-f900e19b-3924-485c-bb47-01b84c559034   1Gi        RWO            ocs-
storagecluster-ceph-rbd   111s
```

3. Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
rbd ls ocs-storagecluster-cephblockpool
```

```
csi-snap-a1e24408-2848-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012
csi-vol-063b982d-2845-11ed-94bd-0a580a830012-temp
csi-vol-5f6e0737-2849-11ed-94bd-0a580a830012
csi-vol-7d67bfad-2842-11ed-94bd-0a580a830012
csi-vol-ed5ce27b-2842-11ed-94bd-0a580a830012
```

```
rbd image-meta ls ocs-storagecluster-cephblockpool/csi-vol-5f6e0737-2849-11ed-94bd-0a580a830012
```

```
There are 4 metadata on this image:

Key                                Value
csi.ceph.com/cluster/name          6cd7a18d-7363-4830-ad5c-f7b96927f026
csi.storage.k8s.io/pv/name         pvc-f900e19b-3924-485c-bb47-01b84c559034
csi.storage.k8s.io/pvc/name        rbd-pvc-restore
csi.storage.k8s.io/pvc/namespace   openshift-storage
```

Verify the metadata for CephFS PVC

Verify that metadata is correctly set on a CephFS PVC by checking the subvolume metadata in the Ceph CLI after creating a persistent volume claim.

Procedure

1. Create a PVC.

```
cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cephfs-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ocs-storagecluster-cephfs
EOF
```

2. Check the status of the PVC.

```
oc get pvc | grep cephfs
```

```
cephfs-pvc                         Bound    pvc-4151128c-86f0-468b-b6e7-5fdfb51ba1b9   1Gi        RWO            ocs-
storagecluster-cephfs     11s
```

3. Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
ceph fs volume ls
```

```
[
    {
        "name": "ocs-storagecluster-cephfilesystem"
    }
]
```

```
ceph fs subvolumegroup ls ocs-storagecluster-cephfilesystem
```

```
[
    {
        "name": "csi"
    }
]
```

```
ceph fs subvolume ls ocs-storagecluster-cephfilesystem --group_name csi
```

```
[
    {
        "name": "csi-vol-25266061-284c-11ed-95e0-0a580a810215"
    }
]
```

```
ceph fs subvolume metadata ls ocs-storagecluster-cephfilesystem csi-vol-25266061-284c-11ed-95e0-0a580a810215 --
group_name=csi --format=json
```

```
    "csi.ceph.com/cluster/name": "6cd7a18d-7363-4830-ad5c-f7b96927f026",
    "csi.storage.k8s.io/pv/name": "pvc-4151128c-86f0-468b-b6e7-5fdfb51ba1b9",
    "csi.storage.k8s.io/pvc/name": "cephfs-pvc",
    "csi.storage.k8s.io/pvc/namespace": "openshift-storage"
}
```

Verify the metadata for CephFS clone

Verify that metadata is correctly set on a CephFS clone by checking the subvolume metadata in the Ceph CLI after creating a PVC clone.

Procedure

1. Create a clone.

```
cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cephfs-pvc-clone
spec:
  storageClassName: ocs-storagecluster-cephfs
  dataSource:
    name: cephfs-pvc
    kind: PersistentVolumeClaim
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
EOF
persistentvolumeclaim/cephfs-pvc-clone created
```

2. Check the status of the clone.

```
oc get pvc | grep cephfs
```

```
cephfs-pvc                          Bound    pvc-4151128c-86f0-468b-b6e7-5fdfb51ba1b9   1Gi        RWO            ocs-
storagecluster-cephfs     9m5s
cephfs-pvc-clone                    Bound    pvc-3d4c4e78-f7d5-456a-aa6e-4da4a05ca4ce   1Gi        RWX            ocs-
storagecluster-cephfs     20s
```

3. Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
ceph fs subvolume ls ocs-storagecluster-cephfilesystem --group_name csi
```

```
[
    {
        "name": "csi-vol-5ea23eb0-284d-11ed-95e0-0a580a810215"
    },
    {
        "name": "csi-vol-25266061-284c-11ed-95e0-0a580a810215"
    }
]
```

```
ceph fs subvolume metadata ls ocs-storagecluster-cephfilesystem csi-vol-5ea23eb0-284d-11ed-95e0-0a580a810215 --
group_name=csi --format=json
```

```
{
    "csi.ceph.com/cluster/name": "6cd7a18d-7363-4830-ad5c-f7b96927f026",
    "csi.storage.k8s.io/pv/name": "pvc-3d4c4e78-f7d5-456a-aa6e-4da4a05ca4ce",
    "csi.storage.k8s.io/pvc/name": "cephfs-pvc-clone",
    "csi.storage.k8s.io/pvc/namespace": "openshift-storage"
}
```

Verify the metadata for CephFS volume snapshot

Verify that metadata is correctly set on a CephFS volume snapshot by checking the subvolume snapshot metadata in the Ceph CLI after creating a volume snapshot.

Procedure

1. Create a volume snapshot.

```
$ cat <<EOF | oc create -f -
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: cephfs-pvc-snapshot
spec:
  volumeSnapshotClassName: ocs-storagecluster-cephfsplugin-snapclass
  source:
    persistentVolumeClaimName: cephfs-pvc
EOF
volumesnapshot.snapshot.storage.k8s.io/cephfs-pvc-snapshot created
```

2. Check the status of the volume snapshot.

```
oc get volumesnapshot
```

```
NAME                  READYTOUSE   SOURCEPVC    SOURCESNAPSHOTCONTENT   RESTORESIZE   SNAPSHOTCLASS
SNAPSHOTCONTENT                                    CREATIONTIME   AGE
cephfs-pvc-snapshot   true         cephfs-pvc                           1Gi           ocs-storagecluster-cephfsplugin-
snapclass   snapcontent-f0f17463-d13b-4e13-b44e-6340bbb3bee0   9s             9s
```

3. Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
ceph fs subvolume snapshot ls ocs-storagecluster-cephfilesystem csi-vol-25266061-284c-11ed-95e0-0a580a810215 --group_name
csi
```

```
[
    {
        "name": "csi-snap-06336f4e-284e-11ed-95e0-0a580a810215"
    }
]
```

```
ceph fs subvolume snapshot metadata ls ocs-storagecluster-cephfilesystem csi-vol-25266061-284c-11ed-95e0-0a580a810215 csi-
snap-06336f4e-284e-11ed-95e0-0a580a810215 --group_name=csi --format=json
```

```
{
    "csi.ceph.com/cluster/name": "6cd7a18d-7363-4830-ad5c-f7b96927f026",
    "csi.storage.k8s.io/volumesnapshot/name": "cephfs-pvc-snapshot",
    "csi.storage.k8s.io/volumesnapshot/namespace": "openshift-storage",
    "csi.storage.k8s.io/volumesnapshotcontent/name": "snapcontent-f0f17463-d13b-4e13-b44e-6340bbb3bee0"
}
```

Verify the metadata of CephFS Restore

Verify that metadata is correctly set on a restored CephFS PVC by checking the subvolume metadata in the Ceph CLI after restoring a volume snapshot.

Procedure

1. Restore a volume snapshot.

```
cat <<EOF | oc create -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cephfs-pvc-restore
spec:
  storageClassName: ocs-storagecluster-cephfs
  dataSource:
    name: cephfs-pvc-snapshot
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
EOF
persistentvolumeclaim/cephfs-pvc-restore created
```

2. Check the status of the restored volume snapshot.

```
oc get pvc | grep cephfs
```

```
cephfs-pvc                          Bound    pvc-4151128c-86f0-468b-b6e7-5fdfb51ba1b9   1Gi        RWO            ocs-
storagecluster-cephfs     29m
cephfs-pvc-clone                    Bound    pvc-3d4c4e78-f7d5-456a-aa6e-4da4a05ca4ce   1Gi        RWX            ocs-
storagecluster-cephfs     20m
cephfs-pvc-restore                  Bound    pvc-43d55ea1-95c0-42c8-8616-4ee70b504445   1Gi        RWX            ocs-
storagecluster-cephfs     21s
```

3. Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
ceph fs subvolume ls ocs-storagecluster-cephfilesystem --group_name csi
```

```
[
    {
        "name": "csi-vol-3536db13-2850-11ed-95e0-0a580a810215"
    },
```

```
    {
        "name": "csi-vol-5ea23eb0-284d-11ed-95e0-0a580a810215"
    },
    {
        "name": "csi-vol-25266061-284c-11ed-95e0-0a580a810215"
    }
]
```

```
ceph fs subvolume metadata ls ocs-storagecluster-cephfilesystem csi-vol-3536db13-2850-11ed-95e0-0a580a810215 --
group_name=csi --format=json
```

```
{
    "csi.ceph.com/cluster/name": "6cd7a18d-7363-4830-ad5c-f7b96927f026",
    "csi.storage.k8s.io/pv/name": "pvc-43d55ea1-95c0-42c8-8616-4ee70b504445",
    "csi.storage.k8s.io/pvc/name": "cephfs-pvc-restore",
    "csi.storage.k8s.io/pvc/namespace": "openshift-storage"
}
```

Alerts

Internal mode cluster alerts are displayed in the dashboard and the OpenShift Container Platform.

- **Setting up alerts**
  For internal Mode clusters, various alerts related to the storage metrics services, storage cluster, disk devices, cluster health, cluster capacity, and so on are displayed in the Block and File, and the object dashboards.

Setting up alerts

For internal Mode clusters, various alerts related to the storage metrics services, storage cluster, disk devices, cluster health, cluster capacity, and so on are displayed in the Block and File, and the object dashboards.

These alerts are not available for external Mode.

Note: It might take a few minutes for alerts to be shown in the alert panel, because only firing alerts are visible in this panel.
You can also view alerts with additional details and customize the display of Alerts in the OpenShift Container Platform.

For more information, see Managing alerts chapter in *OpenShift Container Platform's Monitoring guide*.

Fusion Data Foundation metrics

Reference guide for ceph-exporter, rook, and NooBaa metrics to improve observability and simplify metric interpretation in Fusion Data Foundation.

RBD/Mirroring

`ocs_mirror_daemon_count`
: Mirror Daemon Count

`ocs_pool_mirroring_status`
: Pool Mirroring Status, 0=Disabled, 1=Enabled

`ocs_rbd_client_blocklisted`
: State of the rbd client on a node, 0 = Unblocked, 1 = Blocked

`ocs_rbd_pv_metadata`
: Attributes of Ceph RBD based Persistent Volume

RGW

`ocs_rgw_health_status`
: Health Status of RGW Endpoint, 0=Connected, 1=Progressing & 2=Failure

Storage Client/Provider

`ocs_storage_client_last_heartbeat`
: Unixtime (in sec) of last heartbeat of OCS Storage Client

`ocs_storage_client_operator_version`
: OCS StorageClient encoded Operator Version

`ocs_storage_client_storage_quota_utilization_ratio`
: StorageQuotaUtilizationRatio of ODF Storage Client

`ocs_storage_consumer_metadata`
: Attributes of OCS Storage Consumers

`ocs_storage_provider_operator_version`
: OCS StorageProvider encode Operator Version

StorageCluster

`ocs_storagecluster_failure_domain_count`
: Count of failure domains for StorageCluster with given name and namespace

ocs_storagecluster_kms_connection_status
KMS Connection Status; 0: Connected, 1: Not Connected, 2: KMS not enabled

Prometheus/HTTP handler

promhttp_metric_handler_errors_total
Total number of internal errors encountered by the promhttp metric handler

Ceph Metrics

ceph_AsyncMessenger_Worker_msgr_connection_idle_timeouts
Number of connections closed due to idleness
ceph_AsyncMessenger_Worker_msgr_connection_ready_timeouts
Number of not yet ready connections declared as dead
ceph_blk_kernel_device_bluestore_discard_op
Number of discard ops issued to kernel device
ceph_blk_kernel_device_bluestore_discard_threads
Number of discard threads running
ceph_blk_kernel_device_db_discard_op
Number of discard ops issued to kernel device
ceph_blk_kernel_device_db_discard_threads
Number of discard threads running
ceph_bluefs_alloc_db_max_lat
Max allocation latency for db device
ceph_bluefs_alloc_slow_fallback
Amount of allocations that required fallback to slow/shared device
ceph_bluefs_alloc_slow_max_lat
Max allocation latency for primary/shared device
ceph_bluefs_alloc_slow_size_fallback
Amount of allocations that required fallback to shared device's regular unit size
ceph_bluefs_alloc_unit_db
Allocation unit size (in bytes) for standalone DB device
ceph_bluefs_alloc_unit_slow
Allocation unit size (in bytes) for primary/shared device
ceph_bluefs_alloc_unit_wal
Allocation unit size (in bytes) for standalone WAL device
ceph_bluefs_alloc_wal_max_lat
Max allocation latency for wal device
ceph_bluefs_bytes_written_slow
Bytes written to WAL/SSTs at slow device
ceph_bluefs_bytes_written_sst
Bytes written to SSTs
ceph_bluefs_bytes_written_wal
Bytes written to WAL
ceph_bluefs_compact_lat_count
Average bluefs log compaction latency Count
ceph_bluefs_compact_lat_sum
Average bluefs log compaction latency Total
ceph_bluefs_compact_lock_lat_count
Average lock duration while compacting bluefs log Count
ceph_bluefs_compact_lock_lat_sum
Average lock duration while compacting bluefs log Total
ceph_bluefs_db_alloc_lat_count
Average bluefs db allocate latency Count
ceph_bluefs_db_alloc_lat_sum
Average bluefs db allocate latency Total
ceph_bluefs_db_total_bytes
Total bytes (main db device)
ceph_bluefs_db_used_bytes
Used bytes (main db device)
ceph_bluefs_flush_lat_count
Average bluefs flush latency Count
ceph_bluefs_flush_lat_sum
Average bluefs flush latency Total
ceph_bluefs_fsync_lat_count
Average bluefs fsync latency Count
ceph_bluefs_fsync_lat_sum
Average bluefs fsync latency Total
ceph_bluefs_log_bytes
Size of the metadata log
ceph_bluefs_logged_bytes
Bytes written to the metadata log
ceph_bluefs_max_bytes_db
Maximum bytes allocated from DB
ceph_bluefs_max_bytes_slow
Maximum bytes allocated from SLOW
ceph_bluefs_max_bytes_wal
Maximum bytes allocated from WAL
ceph_bluefs_num_files
File count
ceph_bluefs_read_bytes
Bytes requested in buffered read mode
ceph_bluefs_read_count
buffered read requests processed
ceph_bluefs_read_disk_bytes

Bytes read in buffered mode from disk

ceph_bluefs_read_disk_bytes_db
reads requests going to DB disk

ceph_bluefs_read_disk_bytes_slow
reads requests going to main disk

ceph_bluefs_read_disk_bytes_wal
reads requests going to WAL disk

ceph_bluefs_read_disk_count
buffered reads requests going to disk

ceph_bluefs_read_lat_count
Average bluefs read latency Count

ceph_bluefs_read_lat_sum
Average bluefs read latency Total

ceph_bluefs_read_prefetch_bytes
Bytes requested in prefetch read mode

ceph_bluefs_read_prefetch_count
prefetch read requests processed

ceph_bluefs_read_random_buffer_bytes
Bytes read from prefetch buffer in random read mode

ceph_bluefs_read_random_buffer_count
random read requests processed using prefetch buffer

ceph_bluefs_read_random_bytes
Bytes requested in random read mode

ceph_bluefs_read_random_count
random read requests processed

ceph_bluefs_read_random_disk_bytes
Bytes read from disk in random read mode

ceph_bluefs_read_random_disk_bytes_db
random reads requests going to DB disk

ceph_bluefs_read_random_disk_bytes_slow
random reads requests going to main disk

ceph_bluefs_read_random_disk_bytes_wal
random reads requests going to WAL disk

ceph_bluefs_read_random_disk_count
random reads requests going to disk

ceph_bluefs_read_random_lat_count
Average bluefs read_random latency Count

ceph_bluefs_read_random_lat_sum
Average bluefs read_random latency Total

ceph_bluefs_slow_alloc_lat_count
Average allocation latency for primary/shared device Count

ceph_bluefs_slow_alloc_lat_sum
Average allocation latency for primary/shared device Total

ceph_bluefs_slow_total_bytes
Total bytes (slow device)

ceph_bluefs_slow_used_bytes
Used bytes (slow device)

ceph_bluefs_truncate_lat_count
Average bluefs truncate latency Count

ceph_bluefs_truncate_lat_sum
Average bluefs truncate latency Total

ceph_bluefs_unlink_lat_count
Average bluefs unlink latency Count

ceph_bluefs_unlink_lat_sum
Average bluefs unlink latency Total

ceph_bluefs_wal_alloc_lat_count
Average bluefs wal allocate latency Count

ceph_bluefs_wal_alloc_lat_sum
Average bluefs wal allocate latency Total

ceph_bluefs_wal_total_bytes
Total bytes (wal device)

ceph_bluefs_wal_used_bytes
Used bytes (wal device)

ceph_bluefs_write_bytes
Bytes written

ceph_bluestore_alloc_unit
allocation unit size in bytes

ceph_bluestore_allocated
Sum for allocated bytes

ceph_bluestore_allocator_lat_count
Average bluestore allocator latency Count

ceph_bluestore_allocator_lat_sum
Average bluestore allocator latency Total

ceph_bluestore_clist_lat_count
Average collection listing latency Count

ceph_bluestore_clist_lat_sum
Average collection listing latency Total

ceph_bluestore_compress_lat_count
Average compress latency Count

ceph_bluestore_compress_lat_sum
Average compress latency Total

ceph_bluestore_compressed
Sum for stored compressed bytes

ceph_bluestore_compressed_allocated
Sum for bytes allocated for compressed data

ceph_bluestore_compressed_original
Sum for original bytes that were compressed

ceph_bluestore_csum_lat_count
Average checksum latency Count
ceph_bluestore_csum_lat_sum
Average checksum latency Total
ceph_bluestore_decompress_lat_count
Average decompress latency Count
ceph_bluestore_decompress_lat_sum
Average decompress latency Total
ceph_bluestore_fragmentation_micros
How fragmented bluestore free space is (free extents / max possible number of free extents) * 1000
ceph_bluestore_kv_commit_lat_count
Average kv_thread commit latency Count
ceph_bluestore_kv_commit_lat_sum
Average kv_thread commit latency Total
ceph_bluestore_kv_final_lat_count
Average kv_finalize thread latency Count
ceph_bluestore_kv_final_lat_sum
Average kv_finalize thread latency Total
ceph_bluestore_kv_flush_lat_count
Average kv_thread flush latency Count
ceph_bluestore_kv_flush_lat_sum
Average kv_thread flush latency Total
ceph_bluestore_kv_sync_lat_count
Average kv_sync thread latency Count
ceph_bluestore_kv_sync_lat_sum
Average kv_sync thread latency Total
ceph_bluestore_omap_get_keys_lat_count
Average omap get_keys call latency Count
ceph_bluestore_omap_get_keys_lat_sum
Average omap get_keys call latency Total
ceph_bluestore_omap_get_values_lat_count
Average omap get_values call latency Count
ceph_bluestore_omap_get_values_lat_sum
Average omap get_values call latency Total
ceph_bluestore_omap_lower_bound_lat_count
Average omap iterator lower_bound call latency Count
ceph_bluestore_omap_lower_bound_lat_sum
Average omap iterator lower_bound call latency Total
ceph_bluestore_omap_next_lat_count
Average omap iterator next call latency Count
ceph_bluestore_omap_next_lat_sum
Average omap iterator next call latency Total
ceph_bluestore_omap_seek_to_first_lat_count
Average omap iterator seek_to_first call latency Count
ceph_bluestore_omap_seek_to_first_lat_sum
Average omap iterator seek_to_first call latency Total
ceph_bluestore_omap_upper_bound_lat_count
Average omap iterator upper_bound call latency Count
ceph_bluestore_omap_upper_bound_lat_sum
Average omap iterator upper_bound call latency Total
ceph_bluestore_onode_hits
Count of onode cache lookup hits
ceph_bluestore_onode_misses
Count of onode cache lookup misses
ceph_bluestore_pricache:data_committed_bytes
total bytes committed,
ceph_bluestore_pricache:data_pri0_bytes
bytes allocated to pri0
ceph_bluestore_pricache:data_pri10_bytes
bytes allocated to pri10
ceph_bluestore_pricache:data_pri11_bytes
bytes allocated to pri11
ceph_bluestore_pricache:data_pri1_bytes
bytes allocated to pri1
ceph_bluestore_pricache:data_pri2_bytes
bytes allocated to pri2
ceph_bluestore_pricache:data_pri3_bytes
bytes allocated to pri3
ceph_bluestore_pricache:data_pri4_bytes
bytes allocated to pri4
ceph_bluestore_pricache:data_pri5_bytes
bytes allocated to pri5
ceph_bluestore_pricache:data_pri6_bytes
bytes allocated to pri6
ceph_bluestore_pricache:data_pri7_bytes
bytes allocated to pri7
ceph_bluestore_pricache:data_pri8_bytes
bytes allocated to pri8
ceph_bluestore_pricache:data_pri9_bytes
bytes allocated to pri9
ceph_bluestore_pricache:data_reserved_bytes
bytes reserved for future growth.
ceph_bluestore_pricache:kv_committed_bytes
total bytes committed,
ceph_bluestore_pricache:kv_onode_committed_bytes
total bytes committed,
ceph_bluestore_pricache:kv_onode_pri0_bytes

bytes allocated to pri0

ceph_bluestore_pricache:kv_onode_pri10_bytes
bytes allocated to pri10

ceph_bluestore_pricache:kv_onode_pri11_bytes
bytes allocated to pri11

ceph_bluestore_pricache:kv_onode_pri1_bytes
bytes allocated to pri1

ceph_bluestore_pricache:kv_onode_pri2_bytes
bytes allocated to pri2

ceph_bluestore_pricache:kv_onode_pri3_bytes
bytes allocated to pri3

ceph_bluestore_pricache:kv_onode_pri4_bytes
bytes allocated to pri4

ceph_bluestore_pricache:kv_onode_pri5_bytes
bytes allocated to pri5

ceph_bluestore_pricache:kv_onode_pri6_bytes
bytes allocated to pri6

ceph_bluestore_pricache:kv_onode_pri7_bytes
bytes allocated to pri7

ceph_bluestore_pricache:kv_onode_pri8_bytes
bytes allocated to pri8

ceph_bluestore_pricache:kv_onode_pri9_bytes
bytes allocated to pri9

ceph_bluestore_pricache:kv_onode_reserved_bytes
bytes reserved for future growth.

ceph_bluestore_pricache:kv_pri0_bytes
bytes allocated to pri0

ceph_bluestore_pricache:kv_pri10_bytes
bytes allocated to pri10

ceph_bluestore_pricache:kv_pri11_bytes
bytes allocated to pri11

ceph_bluestore_pricache:kv_pri1_bytes
bytes allocated to pri1

ceph_bluestore_pricache:kv_pri2_bytes
bytes allocated to pri2

ceph_bluestore_pricache:kv_pri3_bytes
bytes allocated to pri3

ceph_bluestore_pricache:kv_pri4_bytes
bytes allocated to pri4

ceph_bluestore_pricache:kv_pri5_bytes
bytes allocated to pri5

ceph_bluestore_pricache:kv_pri6_bytes
bytes allocated to pri6

ceph_bluestore_pricache:kv_pri7_bytes
bytes allocated to pri7

ceph_bluestore_pricache:kv_pri8_bytes
bytes allocated to pri8

ceph_bluestore_pricache:kv_pri9_bytes
bytes allocated to pri9

ceph_bluestore_pricache:kv_reserved_bytes
bytes reserved for future growth.

ceph_bluestore_pricache:meta_committed_bytes
total bytes committed,

ceph_bluestore_pricache:meta_pri0_bytes
bytes allocated to pri0

ceph_bluestore_pricache:meta_pri10_bytes
bytes allocated to pri10

ceph_bluestore_pricache:meta_pri11_bytes
bytes allocated to pri11

ceph_bluestore_pricache:meta_pri1_bytes
bytes allocated to pri1

ceph_bluestore_pricache:meta_pri2_bytes
bytes allocated to pri2

ceph_bluestore_pricache:meta_pri3_bytes
bytes allocated to pri3

ceph_bluestore_pricache:meta_pri4_bytes
bytes allocated to pri4

ceph_bluestore_pricache:meta_pri5_bytes
bytes allocated to pri5

ceph_bluestore_pricache:meta_pri6_bytes
bytes allocated to pri6

ceph_bluestore_pricache:meta_pri7_bytes
bytes allocated to pri7

ceph_bluestore_pricache:meta_pri8_bytes
bytes allocated to pri8

ceph_bluestore_pricache:meta_pri9_bytes
bytes allocated to pri9

ceph_bluestore_pricache:meta_reserved_bytes
bytes reserved for future growth.

ceph_bluestore_pricache_cache_bytes
current memory available for caches.

ceph_bluestore_pricache_heap_bytes
aggregate bytes in use by the heap

ceph_bluestore_pricache_mapped_bytes
total bytes mapped by the process

ceph_bluestore_pricache_target_bytes
target process memory usage in bytes

ceph_bluestore_pricache_unmapped_bytes
: unmapped bytes that the kernel has yet to reclaim

ceph_bluestore_read_lat_count
: Average read latency Count

ceph_bluestore_read_lat_sum
: Average read latency Total

ceph_bluestore_read_onode_meta_lat_count
: Average read onode metadata latency Count

ceph_bluestore_read_onode_meta_lat_sum
: Average read onode metadata latency Total

ceph_bluestore_read_wait_aio_lat_count
: Average read I/O waiting latency Count

ceph_bluestore_read_wait_aio_lat_sum
: Average read I/O waiting latency Total

ceph_bluestore_reads_with_retries
: Read operations that required at least one retry due to failed checksum validation

ceph_bluestore_remove_lat_count
: Average removal latency Count

ceph_bluestore_remove_lat_sum
: Average removal latency Total

ceph_bluestore_slow_aio_wait_count
: Slow op count for aio wait

ceph_bluestore_slow_committed_kv_count
: Slow op count for committed kv

ceph_bluestore_slow_read_onode_meta_count
: Slow op count for read onode meta

ceph_bluestore_slow_read_wait_aio_count
: Slow op count for read wait aio

ceph_bluestore_state_aio_wait_lat_count
: Average aio_wait state latency Count

ceph_bluestore_state_aio_wait_lat_sum
: Average aio_wait state latency Total

ceph_bluestore_state_deferred_aio_wait_lat_count
: Average aio_wait state latency Count

ceph_bluestore_state_deferred_aio_wait_lat_sum
: Average aio_wait state latency Total

ceph_bluestore_state_deferred_cleanup_lat_count
: Average cleanup state latency Count

ceph_bluestore_state_deferred_cleanup_lat_sum
: Average cleanup state latency Total

ceph_bluestore_state_deferred_queued_lat_count
: Average deferred_queued state latency Count

ceph_bluestore_state_deferred_queued_lat_sum
: Average deferred_queued state latency Total

ceph_bluestore_state_done_lat_count
: Average done state latency Count

ceph_bluestore_state_done_lat_sum
: Average done state latency Total

ceph_bluestore_state_finishing_lat_count
: Average finishing state latency Count

ceph_bluestore_state_finishing_lat_sum
: Average finishing state latency Total

ceph_bluestore_state_io_done_lat_count
: Average io_done state latency Count

ceph_bluestore_state_io_done_lat_sum
: Average io_done state latency Total

ceph_bluestore_state_kv_commiting_lat_count
: Average kv_commiting state latency Count

ceph_bluestore_state_kv_commiting_lat_sum
: Average kv_commiting state latency Total

ceph_bluestore_state_kv_done_lat_count
: Average kv_done state latency Count

ceph_bluestore_state_kv_done_lat_sum
: Average kv_done state latency Total

ceph_bluestore_state_kv_queued_lat_count
: Average kv_queued state latency Count

ceph_bluestore_state_kv_queued_lat_sum
: Average kv_queued state latency Total

ceph_bluestore_state_prepare_lat_count
: Average prepare state latency Count

ceph_bluestore_state_prepare_lat_sum
: Average prepare state latency Total

ceph_bluestore_stored
: Sum for stored bytes

ceph_bluestore_truncate_lat_count
: Average truncate latency Count

ceph_bluestore_truncate_lat_sum
: Average truncate latency Total

ceph_bluestore_txc_commit_lat_count
: Average commit latency Count

ceph_bluestore_txc_commit_lat_sum
: Average commit latency Total

ceph_bluestore_txc_submit_lat_count
: Average submit latency Count

ceph_bluestore_txc_submit_lat_sum
: Average submit latency Total

ceph_bluestore_txc_throttle_lat_count

Average submit throttle latency Count

ceph_bluestore_txc_throttle_lat_sum

Average submit throttle latency Total

ceph_daemon_socket_up

Reports the health status of a Ceph daemon, as determined by whether it is able to respond through its admin socket (1 = healthy, 0 = unhealthy).

ceph_exporter_scrape_time

Time spent scraping and transforming perf counters to metrics

ceph_mds_cache_ireq_enqueue_scrub

Internal Request type enqueue scrub

ceph_mds_cache_ireq_exportdir

Internal Request type export dir

ceph_mds_cache_ireq_flush

Internal Request type flush

ceph_mds_cache_ireq_fragmentdir

Internal Request type fragmentdir

ceph_mds_cache_ireq_fragstats

Internal Request type frag stats

ceph_mds_cache_ireq_inodestats

Internal Request type inode stats

ceph_mds_cache_ireq_quiesce_inode

Internal Request type quiesce subvolume inode

ceph_mds_cache_ireq_quiesce_path

Internal Request type quiesce subvolume

ceph_mds_cache_num_recovering_enqueued

Files waiting for recovery

ceph_mds_cache_num_recovering_prioritized

Files waiting for recovery with elevated priority

ceph_mds_cache_num_recovering_processing

Files currently being recovered

ceph_mds_cache_num_strays

Stray dentries

ceph_mds_cache_num_strays_delayed

Stray dentries delayed

ceph_mds_cache_num_strays_enqueuing

Stray dentries enqueuing for purge

ceph_mds_cache_recovery_completed

File recoveries completed

ceph_mds_cache_recovery_started

File recoveries started

ceph_mds_cache_strays_created

Stray dentries created

ceph_mds_cache_strays_enqueued

Stray dentries enqueued for purge

ceph_mds_cache_strays_migrated

Stray dentries migrated

ceph_mds_cache_strays_reintegrated

Stray dentries reintegrated

ceph_mds_caps

Capabilities

ceph_mds_ceph_cap_op_flush_ack

caps truncate notify

ceph_mds_ceph_cap_op_flushsnap_ack

caps truncate notify

ceph_mds_ceph_cap_op_grant

Grant caps

ceph_mds_ceph_cap_op_revoke

Revoke caps

ceph_mds_ceph_cap_op_trunc

caps truncate notify

ceph_mds_client_metrics_num_clients

Numer of client sessions

ceph_mds_dir_commit

Directory commit

ceph_mds_dir_fetch_complete

Fetch complete dirfrag

ceph_mds_dir_fetch_keys

Fetch keys from dirfrag

ceph_mds_dir_merge

Directory merge

ceph_mds_dir_split

Directory split

ceph_mds_exported_inodes

Exported inodes

ceph_mds_forward

Forwarding request

ceph_mds_handle_client_cap_release

Client cap release msg

ceph_mds_handle_client_caps

Client caps msg

ceph_mds_handle_client_caps_dirty

Client dirty caps msg

ceph_mds_handle_inode_file_caps

Inter mds caps msg

ceph_mds_imported_inodes

Imported inodes

ceph_mds_inodes

Inodes

ceph_mds_inodes_expired
: Inodes expired

ceph_mds_inodes_pinned
: Inodes pinned

ceph_mds_inodes_with_caps
: Inodes with capabilities

ceph_mds_load_cent
: Load per cent

ceph_mds_log_ev
: Events

ceph_mds_log_evadd
: Events submitted

ceph_mds_log_evex
: Total expired events

ceph_mds_log_evexd
: Current expired events

ceph_mds_log_evexg
: Expiring events

ceph_mds_log_evlrg
: Large events

ceph_mds_log_evtrm
: Trimmed events

ceph_mds_log_jlat_count
: Journaler flush latency Count

ceph_mds_log_jlat_sum
: Journaler flush latency Total

ceph_mds_log_replayed
: Events replayed

ceph_mds_log_seg
: Segments

ceph_mds_log_segadd
: Segments added

ceph_mds_log_segex
: Total expired segments

ceph_mds_log_segexd
: Current expired segments

ceph_mds_log_segexg
: Expiring segments

ceph_mds_log_segmjr
: Major Segments

ceph_mds_log_segtrm
: Trimmed segments

ceph_mds_mem_cap
: Capabilities

ceph_mds_mem_cap_minus
: Capabilities removed

ceph_mds_mem_cap_plus
: Capabilities added

ceph_mds_mem_dir
: Directories

ceph_mds_mem_dir_minus
: Directories closed

ceph_mds_mem_dir_plus
: Directories opened

ceph_mds_mem_dn
: Dentries

ceph_mds_mem_dn_minus
: Dentries closed

ceph_mds_mem_dn_plus
: Dentries opened

ceph_mds_mem_heap
: Heap size

ceph_mds_mem_ino
: Inodes

ceph_mds_mem_ino_minus
: Inodes closed

ceph_mds_mem_ino_plus
: Inodes opened

ceph_mds_mem_rss
: RSS

ceph_mds_openino_dir_fetch
: OpenIno incomplete directory fetchings

ceph_mds_process_request_cap_release
: Process request cap release

ceph_mds_reply_latency_count
: Reply latency Count

ceph_mds_reply_latency_sum
: Reply latency Total

ceph_mds_request
: Requests

ceph_mds_root_rbytes
: root inode rbytes

ceph_mds_root_rfiles
: root inode rfiles

ceph_mds_root_rsnaps
: root inode rsnaps

ceph_mds_server_cap_acquisition_throttle

Cap acquisition throttle counter

ceph_mds_server_cap_revoke_eviction

Cap Revoke Client Eviction

ceph_mds_server_handle_client_request

Client requests

ceph_mds_server_handle_client_session

Client session messages

ceph_mds_server_handle_peer_request

Peer requests

ceph_mds_server_req_blockdiff_latency_count

Request type file blockdiff latency Count

ceph_mds_server_req_blockdiff_latency_sum

Request type file blockdiff latency Total

ceph_mds_server_req_create_latency_count

Request type create latency Count

ceph_mds_server_req_create_latency_sum

Request type create latency Total

ceph_mds_server_req_getattr_latency_count

Request type get attribute latency Count

ceph_mds_server_req_getattr_latency_sum

Request type get attribute latency Total

ceph_mds_server_req_getfilelock_latency_count

Request type get file lock latency Count

ceph_mds_server_req_getfilelock_latency_sum

Request type get file lock latency Total

ceph_mds_server_req_getvxattr_latency_count

Request type get virtual extended attribute latency Count

ceph_mds_server_req_getvxattr_latency_sum

Request type get virtual extended attribute latency Total

ceph_mds_server_req_link_latency_count

Request type link latency Count

ceph_mds_server_req_link_latency_sum

Request type link latency Total

ceph_mds_server_req_lookup_latency_count

Request type lookup latency Count

ceph_mds_server_req_lookup_latency_sum

Request type lookup latency Total

ceph_mds_server_req_lookuphash_latency_count

Request type lookup hash of inode latency Count

ceph_mds_server_req_lookuphash_latency_sum

Request type lookup hash of inode latency Total

ceph_mds_server_req_lookupino_latency_count

Request type lookup inode latency Count

ceph_mds_server_req_lookupino_latency_sum

Request type lookup inode latency Total

ceph_mds_server_req_lookupname_latency_count

Request type lookup name latency Count

ceph_mds_server_req_lookupname_latency_sum

Request type lookup name latency Total

ceph_mds_server_req_lookupparent_latency_count

Request type lookup parent latency Count

ceph_mds_server_req_lookupparent_latency_sum

Request type lookup parent latency Total

ceph_mds_server_req_lookupsnap_latency_count

Request type lookup snapshot latency Count

ceph_mds_server_req_lookupsnap_latency_sum

Request type lookup snapshot latency Total

ceph_mds_server_req_lssnap_latency_count

Request type list snapshot latency Count

ceph_mds_server_req_lssnap_latency_sum

Request type list snapshot latency Total

ceph_mds_server_req_mkdir_latency_count

Request type make directory latency Count

ceph_mds_server_req_mkdir_latency_sum

Request type make directory latency Total

ceph_mds_server_req_mknod_latency_count

Request type make node latency Count

ceph_mds_server_req_mknod_latency_sum

Request type make node latency Total

ceph_mds_server_req_mksnap_latency_count

Request type make snapshot latency Count

ceph_mds_server_req_mksnap_latency_sum

Request type make snapshot latency Total

ceph_mds_server_req_open_latency_count

Request type open latency Count

ceph_mds_server_req_open_latency_sum

Request type open latency Total

ceph_mds_server_req_readdir_latency_count

Request type read directory latency Count

ceph_mds_server_req_readdir_latency_sum

Request type read directory latency Total

ceph_mds_server_req_rename_latency_count

Request type rename latency Count

ceph_mds_server_req_rename_latency_sum

Request type rename latency Total

ceph_mds_server_req_renamesnap_latency_count

Request type rename snapshot latency Count

ceph_mds_server_req_renamesnap_latency_sum
: Request type rename snapshot latency Total

ceph_mds_server_req_rmdir_latency_count
: Request type remove directory latency Count

ceph_mds_server_req_rmdir_latency_sum
: Request type remove directory latency Total

ceph_mds_server_req_rmsnap_latency_count
: Request type remove snapshot latency Count

ceph_mds_server_req_rmsnap_latency_sum
: Request type remove snapshot latency Total

ceph_mds_server_req_rmxattr_latency_count
: Request type remove extended attribute latency Count

ceph_mds_server_req_rmxattr_latency_sum
: Request type remove extended attribute latency Total

ceph_mds_server_req_setattr_latency_count
: Request type set attribute latency Count

ceph_mds_server_req_setattr_latency_sum
: Request type set attribute latency Total

ceph_mds_server_req_setdirlayout_latency_count
: Request type set directory layout latency Count

ceph_mds_server_req_setdirlayout_latency_sum
: Request type set directory layout latency Total

ceph_mds_server_req_setfilelock_latency_count
: Request type set file lock latency Count

ceph_mds_server_req_setfilelock_latency_sum
: Request type set file lock latency Total

ceph_mds_server_req_setlayout_latency_count
: Request type set file layout latency Count

ceph_mds_server_req_setlayout_latency_sum
: Request type set file layout latency Total

ceph_mds_server_req_setxattr_latency_count
: Request type set extended attribute latency Count

ceph_mds_server_req_setxattr_latency_sum
: Request type set extended attribute latency Total

ceph_mds_server_req_snapdiff_latency_count
: Request type snapshot difference latency Count

ceph_mds_server_req_snapdiff_latency_sum
: Request type snapshot difference latency Total

ceph_mds_server_req_symlink_latency_count
: Request type symbolic link latency Count

ceph_mds_server_req_symlink_latency_sum
: Request type symbolic link latency Total

ceph_mds_server_req_unlink_latency_count
: Request type unlink latency Count

ceph_mds_server_req_unlink_latency_sum
: Request type unlink latency Total

ceph_mds_sessions_average_load
: Average Load

ceph_mds_sessions_avg_session_uptime
: Average session uptime

ceph_mds_sessions_mdthresh_evicted
: Sessions evicted on reaching metadata threshold

ceph_mds_sessions_session_add
: Sessions added

ceph_mds_sessions_session_count
: Session count

ceph_mds_sessions_session_remove
: Sessions removed

ceph_mds_sessions_sessions_open
: Sessions currently open

ceph_mds_sessions_sessions_stale
: Sessions currently stale

ceph_mds_sessions_total_load
: Total Load

ceph_mds_slow_reply
: Slow replies

ceph_mds_subtrees
: Subtrees

ceph_mon_election_call
: Elections started

ceph_mon_election_lose
: Elections lost

ceph_mon_election_win
: Elections won

ceph_mon_num_elections
: Elections participated in

ceph_mon_num_sessions
: Open sessions

ceph_mon_session_add
: Created sessions

ceph_mon_session_rm
: Removed sessions

ceph_mon_session_trim
: Trimmed sessions

ceph_objecter_op_active
: Operations active

ceph_objecter_op_r

Read operations
ceph_objecter_op_rmw
Read-modify-write operations
ceph_objecter_op_w
Write operations
ceph_osd_numpg
Placement groups
ceph_osd_numpg_removing
Placement groups queued for local deletion
ceph_osd_op
Client operations
ceph_osd_op_before_queue_op_lat_count
Latency of IO before calling queue(before really queue into ShardedOpWq) Count
ceph_osd_op_before_queue_op_lat_sum
Latency of IO before calling queue(before really queue into ShardedOpWq) Total
ceph_osd_op_delayed_degraded
Count of ops delayed due to target object being degraded
ceph_osd_op_delayed_unreadable
Count of ops delayed due to target object being unreadable
ceph_osd_op_in_bytes
Client operations total write size
ceph_osd_op_latency_count
Latency of client operations (including queue time) Count
ceph_osd_op_latency_sum
Latency of client operations (including queue time) Total
ceph_osd_op_out_bytes
Client operations total read size
ceph_osd_op_prepare_latency_count
Latency of client operations (excluding queue time and wait for finished) Count
ceph_osd_op_prepare_latency_sum
Latency of client operations (excluding queue time and wait for finished) Total
ceph_osd_op_process_latency_count
Latency of client operations (excluding queue time) Count
ceph_osd_op_process_latency_sum
Latency of client operations (excluding queue time) Total
ceph_osd_op_r
Client read operations
ceph_osd_op_r_latency_count
Latency of read operation (including queue time) Count
ceph_osd_op_r_latency_sum
Latency of read operation (including queue time) Total
ceph_osd_op_r_out_bytes
Client data read
ceph_osd_op_r_prepare_latency_count
Latency of read operations (excluding queue time and wait for finished) Count
ceph_osd_op_r_prepare_latency_sum
Latency of read operations (excluding queue time and wait for finished) Total
ceph_osd_op_r_process_latency_count
Latency of read operation (excluding queue time) Count
ceph_osd_op_r_process_latency_sum
Latency of read operation (excluding queue time) Total
ceph_osd_op_rw
Client read-modify-write operations
ceph_osd_op_rw_in_bytes
Client read-modify-write operations write in
ceph_osd_op_rw_latency_count
Latency of read-modify-write operation (including queue time) Count
ceph_osd_op_rw_latency_sum
Latency of read-modify-write operation (including queue time) Total
ceph_osd_op_rw_out_bytes
Client read-modify-write operations read out
ceph_osd_op_rw_prepare_latency_count
Latency of read-modify-write operations (excluding queue time and wait for finished) Count
ceph_osd_op_rw_prepare_latency_sum
Latency of read-modify-write operations (excluding queue time and wait for finished) Total
ceph_osd_op_rw_process_latency_count
Latency of read-modify-write operation (excluding queue time) Count
ceph_osd_op_rw_process_latency_sum
Latency of read-modify-write operation (excluding queue time) Total
ceph_osd_op_w
Client write operations
ceph_osd_op_w_in_bytes
Client data written
ceph_osd_op_w_latency_count
Latency of write operation (including queue time) Count
ceph_osd_op_w_latency_sum
Latency of write operation (including queue time) Total
ceph_osd_op_w_prepare_latency_count
Latency of write operations (excluding queue time and wait for finished) Count
ceph_osd_op_w_prepare_latency_sum
Latency of write operations (excluding queue time and wait for finished) Total
ceph_osd_op_w_process_latency_count
Latency of write operation (excluding queue time) Count
ceph_osd_op_w_process_latency_sum
Latency of write operation (excluding queue time) Total
ceph_osd_op_wip
Replication operations currently being processed (primary)

ceph_osd_recovery_bytes
: recovery bytes

ceph_osd_recovery_ops
: Started recovery operations

ceph_osd_scrub_dp_ec_chunk_busy
: chunk busy during scrubs

ceph_osd_scrub_dp_ec_chunk_selected
: chunk selection during scrubs

ceph_osd_scrub_dp_ec_failed_reservations_elapsed_count
: time for scrub reservation to fail Count

ceph_osd_scrub_dp_ec_failed_reservations_elapsed_sum
: time for scrub reservation to fail Total

ceph_osd_scrub_dp_ec_failed_scrubs
: failed scrubs count

ceph_osd_scrub_dp_ec_failed_scrubs_elapsed_count
: time to scrub failure Count

ceph_osd_scrub_dp_ec_failed_scrubs_elapsed_sum
: time to scrub failure Total

ceph_osd_scrub_dp_ec_locked_object
: waiting on locked object events

ceph_osd_scrub_dp_ec_num_scrubs_past_reservation
: scrubs count

ceph_osd_scrub_dp_ec_num_scrubs_started
: scrubs attempted count

ceph_osd_scrub_dp_ec_preemptions
: preemptions on scrubs

ceph_osd_scrub_dp_ec_replicas_in_reservation
: number of replicas in reservation

ceph_osd_scrub_dp_ec_reservation_process_aborted
: scrub reservation was aborted

ceph_osd_scrub_dp_ec_reservation_process_failure
: scrub reservation failed due to replica denial

ceph_osd_scrub_dp_ec_reservation_process_skipped
: scrub reservation skipped for high priority scrub

ceph_osd_scrub_dp_ec_scrub_reservations_completed
: successfully completed reservation processes

ceph_osd_scrub_dp_ec_successful_reservations_elapsed_count
: time to scrub reservation completion Count

ceph_osd_scrub_dp_ec_successful_reservations_elapsed_sum
: time to scrub reservation completion Total

ceph_osd_scrub_dp_ec_successful_scrubs
: successful scrubs count

ceph_osd_scrub_dp_ec_successful_scrubs_elapsed_count
: time to scrub completion Count

ceph_osd_scrub_dp_ec_successful_scrubs_elapsed_sum
: time to scrub completion Total

ceph_osd_scrub_dp_ec_write_blocked_by_scrub
: write blocked by scrub

ceph_osd_scrub_dp_repl_chunk_busy
: chunk busy during scrubs

ceph_osd_scrub_dp_repl_chunk_selected
: chunk selection during scrubs

ceph_osd_scrub_dp_repl_failed_reservations_elapsed_count
: time for scrub reservation to fail Count

ceph_osd_scrub_dp_repl_failed_reservations_elapsed_sum
: time for scrub reservation to fail Total

ceph_osd_scrub_dp_repl_failed_scrubs
: failed scrubs count

ceph_osd_scrub_dp_repl_failed_scrubs_elapsed_count
: time to scrub failure Count

ceph_osd_scrub_dp_repl_failed_scrubs_elapsed_sum
: time to scrub failure Total

ceph_osd_scrub_dp_repl_locked_object
: waiting on locked object events

ceph_osd_scrub_dp_repl_num_scrubs_past_reservation
: scrubs count

ceph_osd_scrub_dp_repl_num_scrubs_started
: scrubs attempted count

ceph_osd_scrub_dp_repl_preemptions
: preemptions on scrubs

ceph_osd_scrub_dp_repl_replicas_in_reservation
: number of replicas in reservation

ceph_osd_scrub_dp_repl_reservation_process_aborted
: scrub reservation was aborted

ceph_osd_scrub_dp_repl_reservation_process_failure
: scrub reservation failed due to replica denial

ceph_osd_scrub_dp_repl_reservation_process_skipped
: scrub reservation skipped for high priority scrub

ceph_osd_scrub_dp_repl_scrub_reservations_completed
: successfully completed reservation processes

ceph_osd_scrub_dp_repl_successful_reservations_elapsed_count
: time to scrub reservation completion Count

ceph_osd_scrub_dp_repl_successful_reservations_elapsed_sum
: time to scrub reservation completion Total

ceph_osd_scrub_dp_repl_successful_scrubs
: successful scrubs count

ceph_osd_scrub_dp_repl_successful_scrubs_elapsed_count

time to scrub completion Count

ceph_osd_scrub_dp_repl_successful_scrubs_elapsed_sum
time to scrub completion Total

ceph_osd_scrub_dp_repl_write_blocked_by_scrub
write blocked by scrub

ceph_osd_scrub_sh_ec_chunk_busy
chunk busy during scrubs

ceph_osd_scrub_sh_ec_chunk_selected
chunk selection during scrubs

ceph_osd_scrub_sh_ec_failed_reservations_elapsed_count
time for scrub reservation to fail Count

ceph_osd_scrub_sh_ec_failed_reservations_elapsed_sum
time for scrub reservation to fail Total

ceph_osd_scrub_sh_ec_failed_scrubs
failed scrubs count

ceph_osd_scrub_sh_ec_failed_scrubs_elapsed_count
time to scrub failure Count

ceph_osd_scrub_sh_ec_failed_scrubs_elapsed_sum
time to scrub failure Total

ceph_osd_scrub_sh_ec_locked_object
waiting on locked object events

ceph_osd_scrub_sh_ec_num_scrubs_past_reservation
scrubs count

ceph_osd_scrub_sh_ec_num_scrubs_started
scrubs attempted count

ceph_osd_scrub_sh_ec_preemptions
preemptions on scrubs

ceph_osd_scrub_sh_ec_replicas_in_reservation
number of replicas in reservation

ceph_osd_scrub_sh_ec_reservation_process_aborted
scrub reservation was aborted

ceph_osd_scrub_sh_ec_reservation_process_failure
scrub reservation failed due to replica denial

ceph_osd_scrub_sh_ec_reservation_process_skipped
scrub reservation skipped for high priority scrub

ceph_osd_scrub_sh_ec_scrub_reservations_completed
successfully completed reservation processes

ceph_osd_scrub_sh_ec_successful_reservations_elapsed_count
time to scrub reservation completion Count

ceph_osd_scrub_sh_ec_successful_reservations_elapsed_sum
time to scrub reservation completion Total

ceph_osd_scrub_sh_ec_successful_scrubs
successful scrubs count

ceph_osd_scrub_sh_ec_successful_scrubs_elapsed_count
time to scrub completion Count

ceph_osd_scrub_sh_ec_successful_scrubs_elapsed_sum
time to scrub completion Total

ceph_osd_scrub_sh_ec_write_blocked_by_scrub
write blocked by scrub

ceph_osd_scrub_sh_repl_chunk_busy
chunk busy during scrubs

ceph_osd_scrub_sh_repl_chunk_selected
chunk selection during scrubs

ceph_osd_scrub_sh_repl_failed_reservations_elapsed_count
time for scrub reservation to fail Count

ceph_osd_scrub_sh_repl_failed_reservations_elapsed_sum
time for scrub reservation to fail Total

ceph_osd_scrub_sh_repl_failed_scrubs
failed scrubs count

ceph_osd_scrub_sh_repl_failed_scrubs_elapsed_count
time to scrub failure Count

ceph_osd_scrub_sh_repl_failed_scrubs_elapsed_sum
time to scrub failure Total

ceph_osd_scrub_sh_repl_locked_object
waiting on locked object events

ceph_osd_scrub_sh_repl_num_scrubs_past_reservation
scrubs count

ceph_osd_scrub_sh_repl_num_scrubs_started
scrubs attempted count

ceph_osd_scrub_sh_repl_preemptions
preemptions on scrubs

ceph_osd_scrub_sh_repl_replicas_in_reservation
number of replicas in reservation

ceph_osd_scrub_sh_repl_reservation_process_aborted
scrub reservation was aborted

ceph_osd_scrub_sh_repl_reservation_process_failure
scrub reservation failed due to replica denial

ceph_osd_scrub_sh_repl_reservation_process_skipped
scrub reservation skipped for high priority scrub

ceph_osd_scrub_sh_repl_scrub_reservations_completed
successfully completed reservation processes

ceph_osd_scrub_sh_repl_successful_reservations_elapsed_count
time to scrub reservation completion Count

ceph_osd_scrub_sh_repl_successful_reservations_elapsed_sum
time to scrub reservation completion Total

ceph_osd_scrub_sh_repl_successful_scrubs
successful scrubs count

ceph_osd_scrub_sh_repl_successful_scrubs_elapsed_count
: time to scrub completion Count

ceph_osd_scrub_sh_repl_successful_scrubs_elapsed_sum
: time to scrub completion Total

ceph_osd_scrub_sh_repl_write_blocked_by_scrub
: write blocked by scrub

ceph_osd_stat_bytes
: OSD size

ceph_osd_stat_bytes_used
: Used space

ceph_paxos_accept_timeout
: Accept timeouts

ceph_paxos_begin
: Started and handled begins

ceph_paxos_begin_bytes_count
: Data in transaction on begin Count

ceph_paxos_begin_bytes_sum
: Data in transaction on begin Total

ceph_paxos_begin_keys_count
: Keys in transaction on begin Count

ceph_paxos_begin_keys_sum
: Keys in transaction on begin Total

ceph_paxos_begin_latency_count
: Latency of begin operation Count

ceph_paxos_begin_latency_sum
: Latency of begin operation Total

ceph_paxos_collect
: Peon collects

ceph_paxos_collect_bytes_count
: Data in transaction on peon collect Count

ceph_paxos_collect_bytes_sum
: Data in transaction on peon collect Total

ceph_paxos_collect_keys_count
: Keys in transaction on peon collect Count

ceph_paxos_collect_keys_sum
: Keys in transaction on peon collect Total

ceph_paxos_collect_latency_count
: Peon collect latency Count

ceph_paxos_collect_latency_sum
: Peon collect latency Total

ceph_paxos_collect_timeout
: Collect timeouts

ceph_paxos_collect_uncommitted
: Uncommitted values in started and handled collects

ceph_paxos_commit
: Commits

ceph_paxos_commit_bytes_count
: Data in transaction on commit Count

ceph_paxos_commit_bytes_sum
: Data in transaction on commit Total

ceph_paxos_commit_keys_count
: Keys in transaction on commit Count

ceph_paxos_commit_keys_sum
: Keys in transaction on commit Total

ceph_paxos_commit_latency_count
: Commit latency Count

ceph_paxos_commit_latency_sum
: Commit latency Total

ceph_paxos_lease_ack_timeout
: Lease acknowledgement timeouts

ceph_paxos_lease_timeout
: Lease timeouts

ceph_paxos_new_pn
: New proposal number queries

ceph_paxos_new_pn_latency_count
: New proposal number getting latency Count

ceph_paxos_new_pn_latency_sum
: New proposal number getting latency Total

ceph_paxos_refresh
: Refreshes

ceph_paxos_refresh_latency_count
: Refresh latency Count

ceph_paxos_refresh_latency_sum
: Refresh latency Total

ceph_paxos_restart
: Restarts

ceph_paxos_share_state
: Sharings of state

ceph_paxos_share_state_bytes_count
: Data in shared state Count

ceph_paxos_share_state_bytes_sum
: Data in shared state Total

ceph_paxos_share_state_keys_count
: Keys in shared state Count

ceph_paxos_share_state_keys_sum
: Keys in shared state Total

ceph_paxos_start_leader

Starts in leader role

ceph_paxos_start_peon
Starts in peon role

ceph_paxos_store_state
Store a shared state on disk

ceph_paxos_store_state_bytes_count
Data in transaction in stored state Count

ceph_paxos_store_state_bytes_sum
Data in transaction in stored state Total

ceph_paxos_store_state_keys_count
Keys in transaction in stored state Count

ceph_paxos_store_state_keys_sum
Keys in transaction in stored state Total

ceph_paxos_store_state_latency_count
Storing state latency Count

ceph_paxos_store_state_latency_sum
Storing state latency Total

ceph_prioritycache:full_committed_bytes
total bytes committed,

ceph_prioritycache:full_pri0_bytes
bytes allocated to pri0

ceph_prioritycache:full_pri10_bytes
bytes allocated to pri10

ceph_prioritycache:full_pri11_bytes
bytes allocated to pri11

ceph_prioritycache:full_pri1_bytes
bytes allocated to pri1

ceph_prioritycache:full_pri2_bytes
bytes allocated to pri2

ceph_prioritycache:full_pri3_bytes
bytes allocated to pri3

ceph_prioritycache:full_pri4_bytes
bytes allocated to pri4

ceph_prioritycache:full_pri5_bytes
bytes allocated to pri5

ceph_prioritycache:full_pri6_bytes
bytes allocated to pri6

ceph_prioritycache:full_pri7_bytes
bytes allocated to pri7

ceph_prioritycache:full_pri8_bytes
bytes allocated to pri8

ceph_prioritycache:full_pri9_bytes
bytes allocated to pri9

ceph_prioritycache:full_reserved_bytes
bytes reserved for future growth.

ceph_prioritycache:inc_committed_bytes
total bytes committed,

ceph_prioritycache:inc_pri0_bytes
bytes allocated to pri0

ceph_prioritycache:inc_pri10_bytes
bytes allocated to pri10

ceph_prioritycache:inc_pri11_bytes
bytes allocated to pri11

ceph_prioritycache:inc_pri1_bytes
bytes allocated to pri1

ceph_prioritycache:inc_pri2_bytes
bytes allocated to pri2

ceph_prioritycache:inc_pri3_bytes
bytes allocated to pri3

ceph_prioritycache:inc_pri4_bytes
bytes allocated to pri4

ceph_prioritycache:inc_pri5_bytes
bytes allocated to pri5

ceph_prioritycache:inc_pri6_bytes
bytes allocated to pri6

ceph_prioritycache:inc_pri7_bytes
bytes allocated to pri7

ceph_prioritycache:inc_pri8_bytes
bytes allocated to pri8

ceph_prioritycache:inc_pri9_bytes
bytes allocated to pri9

ceph_prioritycache:inc_reserved_bytes
bytes reserved for future growth.

ceph_prioritycache:kv_committed_bytes
total bytes committed,

ceph_prioritycache:kv_pri0_bytes
bytes allocated to pri0

ceph_prioritycache:kv_pri10_bytes
bytes allocated to pri10

ceph_prioritycache:kv_pri11_bytes
bytes allocated to pri11

ceph_prioritycache:kv_pri1_bytes
bytes allocated to pri1

ceph_prioritycache:kv_pri2_bytes
bytes allocated to pri2

ceph_prioritycache:kv_pri3_bytes
bytes allocated to pri3

ceph_prioritycache:kv_pri4_bytes
bytes allocated to pri4
ceph_prioritycache:kv_pri5_bytes
bytes allocated to pri5
ceph_prioritycache:kv_pri6_bytes
bytes allocated to pri6
ceph_prioritycache:kv_pri7_bytes
bytes allocated to pri7
ceph_prioritycache:kv_pri8_bytes
bytes allocated to pri8
ceph_prioritycache:kv_pri9_bytes
bytes allocated to pri9
ceph_prioritycache:kv_reserved_bytes
bytes reserved for future growth.
ceph_prioritycache_cache_bytes
current memory available for caches.
ceph_prioritycache_heap_bytes
aggregate bytes in use by the heap
ceph_prioritycache_mapped_bytes
total bytes mapped by the process
ceph_prioritycache_target_bytes
target process memory usage in bytes
ceph_prioritycache_unmapped_bytes
unmapped bytes that the kernel has yet to reclaim
ceph_purge_queue_pq_executed
Purge queue tasks executed
ceph_purge_queue_pq_executed_ops
Purge queue ops executed
ceph_purge_queue_pq_executing
Purge queue tasks in flight
ceph_purge_queue_pq_executing_high_water
Maximum number of executing file purges
ceph_purge_queue_pq_executing_ops
Purge queue ops in flight
ceph_purge_queue_pq_executing_ops_high_water
Maximum number of executing file purge ops
ceph_purge_queue_pq_item_in_journal
Purge item left in journal
ceph_rocksdb_compact
Compactions
ceph_rocksdb_compact_completed
Completed compactions
ceph_rocksdb_compact_lasted
Last completed compaction duration
ceph_rocksdb_compact_queue_len
Length of compaction queue
ceph_rocksdb_compact_queue_merge
Mergings of ranges in compaction queue
ceph_rocksdb_compact_running
Running compactions
ceph_rocksdb_get_latency_count
Get latency Count
ceph_rocksdb_get_latency_sum
Get latency Total
ceph_rocksdb_rocksdb_write_delay_time_count
Rocksdb write delay time Count
ceph_rocksdb_rocksdb_write_delay_time_sum
Rocksdb write delay time Total
ceph_rocksdb_rocksdb_write_memtable_time_count
Rocksdb write memtable time Count
ceph_rocksdb_rocksdb_write_memtable_time_sum
Rocksdb write memtable time Total
ceph_rocksdb_rocksdb_write_pre_and_post_time_count
total time spent on writing a record, excluding write process Count
ceph_rocksdb_rocksdb_write_pre_and_post_time_sum
total time spent on writing a record, excluding write process Total
ceph_rocksdb_rocksdb_write_wal_time_count
Rocksdb write wal time Count
ceph_rocksdb_rocksdb_write_wal_time_sum
Rocksdb write wal time Total
ceph_rocksdb_submit_latency_count
Submit Latency Count
ceph_rocksdb_submit_latency_sum
Submit Latency Total
ceph_rocksdb_submit_sync_latency_count
Submit Sync Latency Count
ceph_rocksdb_submit_sync_latency_sum
Submit Sync Latency Total

NooBaa

NooBaa_Endpoint_process_cpu_user_seconds_total
Total user CPU time spent in seconds.
NooBaa_Endpoint_process_cpu_system_seconds_total
Total system CPU time spent in seconds.
NooBaa_Endpoint_process_cpu_seconds_total

Total user and system CPU time spent in seconds.

`NooBaa_Endpoint_process_start_time_seconds`
Start time of the process since unix epoch in seconds.

`NooBaa_Endpoint_process_resident_memory_bytes`
Resident memory size in bytes.

`NooBaa_Endpoint_process_virtual_memory_bytes`
Virtual memory size in bytes.

`NooBaa_Endpoint_process_heap_bytes`
Process heap size in bytes.

`NooBaa_Endpoint_process_open_fds`
Number of open file descriptors.

`NooBaa_Endpoint_process_max_fds`
Maximum number of open file descriptors.

`NooBaa_Endpoint_nodejs_eventloop_lag_seconds`
Lag of event loop in seconds.

`NooBaa_Endpoint_nodejs_eventloop_lag_min_seconds`
The minimum recorded event loop delay.

`NooBaa_Endpoint_nodejs_eventloop_lag_max_seconds`
The maximum recorded event loop delay.

`NooBaa_Endpoint_nodejs_eventloop_lag_mean_seconds`
The mean of the recorded event loop delays.

`NooBaa_Endpoint_nodejs_eventloop_lag_stddev_seconds`
The standard deviation of the recorded event loop delays.

`NooBaa_Endpoint_nodejs_eventloop_lag_p50_seconds`
The 50th percentile of the recorded event loop delays.

`NooBaa_Endpoint_nodejs_eventloop_lag_p90_seconds`
The 90th percentile of the recorded event loop delays.

`NooBaa_Endpoint_nodejs_eventloop_lag_p99_seconds`
The 99th percentile of the recorded event loop delays.

`NooBaa_Endpoint_nodejs_active_resources`
Number of active resources that are currently keeping the event loop alive, grouped by async resource type.

`NooBaa_Endpoint_nodejs_active_resources_total`
Total number of active resources.

`NooBaa_Endpoint_nodejs_active_handles`
Number of active libuv handles grouped by handle type. Every handle type is C++ class name.

`NooBaa_Endpoint_nodejs_active_handles_total`
Total number of active handles.

`NooBaa_Endpoint_nodejs_active_requests`
Number of active libuv requests grouped by request type. Every request type is C++ class name.

`NooBaa_Endpoint_nodejs_active_requests_total`
Total number of active requests.

`NooBaa_Endpoint_nodejs_heap_size_total_bytes`
Process heap size from Node.js in bytes.

`NooBaa_Endpoint_nodejs_heap_size_used_bytes`
Process heap size used from Node.js in bytes.

`NooBaa_Endpoint_nodejs_external_memory_bytes`
Node.js external memory size in bytes.

`NooBaa_Endpoint_nodejs_heap_space_size_total_bytes`
Process heap space size total from Node.js in bytes.

`NooBaa_Endpoint_nodejs_heap_space_size_used_bytes`
Process heap space size used from Node.js in bytes.

`NooBaa_Endpoint_nodejs_heap_space_size_available_bytes`
Process heap space size available from Node.js in bytes.

`NooBaa_Endpoint_nodejs_version_info`
Node.js version info.

`NooBaa_Endpoint_nodejs_gc_duration_seconds`
Garbage collection duration by kind, one of major, minor, incremental or weakcb.

`NooBaa_Endpoint_hub_read_bytes`
hub read bytes in namespace cache bucket

`NooBaa_Endpoint_hub_write_bytes`
hub write bytes in namespace cache bucket

`NooBaa_Endpoint_cache_read_bytes`
Cache read bytes in namespace cache bucket

`NooBaa_Endpoint_cache_write_bytes`
Cache write bytes in namespace cache bucket

`NooBaa_Endpoint_cache_object_read_count`
Counter on entire object reads in namespace cache bucket

`NooBaa_Endpoint_cache_object_read_miss_count`
Counter on entire object read miss in namespace cache bucket

`NooBaa_Endpoint_cache_object_read_hit_count`
Counter on entire object read hit in namespace cache bucket

`NooBaa_Endpoint_cache_range_read_count`
Counter on range reads in namespace cache bucket

`NooBaa_Endpoint_cache_range_read_miss_count`
Counter on range read miss in namespace cache bucket

`NooBaa_Endpoint_cache_range_read_hit_count`
Counter on range read hit in namespace cache bucket

`NooBaa_Endpoint_hub_read_latency`
hub read latency in namespace cache bucket

`NooBaa_Endpoint_hub_write_latency`
hub write latency in namespace cache bucket

`NooBaa_Endpoint_cache_read_latency`
Cache read latency in namespace cache bucket

`NooBaa_Endpoint_cache_write_latency`
Cache write latency in namespace cache bucket

`NooBaa_Endpoint_semaphore_waiting_value`
Namespace semaphore waiting value

NooBaa_Endpoint_semaphore_waiting_time
Namespace semaphore waiting time
NooBaa_Endpoint_semaphore_waiting_queue
Namespace semaphore waiting queue size
NooBaa_Endpoint_semaphore_value
Namespace semaphore value
NooBaa_Endpoint_fork_counter
Counter on number of fork hit
NooBaa_WebServer_process_cpu_user_seconds_total
Total user CPU time spent in seconds.
NooBaa_WebServer_process_cpu_system_seconds_total
Total system CPU time spent in seconds.
NooBaa_WebServer_process_cpu_seconds_total
Total user and system CPU time spent in seconds.
NooBaa_WebServer_process_start_time_seconds
Start time of the process since unix epoch in seconds.
NooBaa_WebServer_process_resident_memory_bytes
Resident memory size in bytes.
NooBaa_WebServer_process_virtual_memory_bytes
Virtual memory size in bytes.
NooBaa_WebServer_process_heap_bytes
Process heap size in bytes.
NooBaa_WebServer_process_open_fds
Number of open file descriptors.
NooBaa_WebServer_process_max_fds
Maximum number of open file descriptors.
NooBaa_WebServer_nodejs_eventloop_lag_seconds
Lag of event loop in seconds.
NooBaa_WebServer_nodejs_eventloop_lag_min_seconds
The minimum recorded event loop delay.
NooBaa_WebServer_nodejs_eventloop_lag_max_seconds
The maximum recorded event loop delay.
NooBaa_WebServer_nodejs_eventloop_lag_mean_seconds
The mean of the recorded event loop delays.
NooBaa_WebServer_nodejs_eventloop_lag_stddev_seconds
The standard deviation of the recorded event loop delays.
NooBaa_WebServer_nodejs_eventloop_lag_p50_seconds
The 50th percentile of the recorded event loop delays.
NooBaa_WebServer_nodejs_eventloop_lag_p90_seconds
The 90th percentile of the recorded event loop delays.
NooBaa_WebServer_nodejs_eventloop_lag_p99_seconds
The 99th percentile of the recorded event loop delays.
NooBaa_WebServer_nodejs_active_resources
Number of active resources that are currently keeping the event loop alive, grouped by async resource type.
NooBaa_WebServer_nodejs_active_resources_total
Total number of active resources.
NooBaa_WebServer_nodejs_active_handles
Number of active libuv handles grouped by handle type. Every handle type is C++ class name.
NooBaa_WebServer_nodejs_active_handles_total
Total number of active handles.
NooBaa_WebServer_nodejs_active_requests
Number of active libuv requests grouped by request type. Every request type is C++ class name.
NooBaa_WebServer_nodejs_active_requests_total
Total number of active requests.
NooBaa_WebServer_nodejs_heap_size_total_bytes
Process heap size from Node.js in bytes.
NooBaa_WebServer_nodejs_heap_size_used_bytes
Process heap size used from Node.js in bytes.
NooBaa_WebServer_nodejs_external_memory_bytes
Node.js external memory size in bytes.
NooBaa_WebServer_nodejs_heap_space_size_total_bytes
Process heap space size total from Node.js in bytes.
NooBaa_WebServer_nodejs_heap_space_size_used_bytes
Process heap space size used from Node.js in bytes.
NooBaa_WebServer_nodejs_heap_space_size_available_bytes
Process heap space size available from Node.js in bytes.
NooBaa_WebServer_nodejs_version_info
Node.js version info.
NooBaa_WebServer_nodejs_gc_duration_seconds
Garbage collection duration by kind, one of major, minor, incremental or weakcb.
NooBaa_BGWorkers_process_cpu_user_seconds_total
Total user CPU time spent in seconds.
NooBaa_BGWorkers_process_cpu_system_seconds_total
Total system CPU time spent in seconds.
NooBaa_BGWorkers_process_cpu_seconds_total
Total user and system CPU time spent in seconds.
NooBaa_BGWorkers_process_start_time_seconds
Start time of the process since unix epoch in seconds.
NooBaa_BGWorkers_process_resident_memory_bytes
Resident memory size in bytes.
NooBaa_BGWorkers_process_virtual_memory_bytes
Virtual memory size in bytes.
NooBaa_BGWorkers_process_heap_bytes
Process heap size in bytes.
NooBaa_BGWorkers_process_open_fds
Number of open file descriptors.
NooBaa_BGWorkers_process_max_fds

Maximum number of open file descriptors.

NooBaa_BGWorkers_nodejs_eventloop_lag_seconds
Lag of event loop in seconds.

NooBaa_BGWorkers_nodejs_eventloop_lag_min_seconds
The minimum recorded event loop delay.

NooBaa_BGWorkers_nodejs_eventloop_lag_max_seconds
The maximum recorded event loop delay.

NooBaa_BGWorkers_nodejs_eventloop_lag_mean_seconds
The mean of the recorded event loop delays.

NooBaa_BGWorkers_nodejs_eventloop_lag_stddev_seconds
The standard deviation of the recorded event loop delays.

NooBaa_BGWorkers_nodejs_eventloop_lag_p50_seconds
The 50th percentile of the recorded event loop delays.

NooBaa_BGWorkers_nodejs_eventloop_lag_p90_seconds
The 90th percentile of the recorded event loop delays.

NooBaa_BGWorkers_nodejs_eventloop_lag_p99_seconds
The 99th percentile of the recorded event loop delays.

NooBaa_BGWorkers_nodejs_active_resources
Number of active resources that are currently keeping the event loop alive, grouped by async resource type.

NooBaa_BGWorkers_nodejs_active_resources_total
Total number of active resources.

NooBaa_BGWorkers_nodejs_active_handles
Number of active libuv handles grouped by handle type. Every handle type is C++ class name.

NooBaa_BGWorkers_nodejs_active_handles_total
Total number of active handles.

NooBaa_BGWorkers_nodejs_active_requests
Number of active libuv requests grouped by request type. Every request type is C++ class name.

NooBaa_BGWorkers_nodejs_active_requests_total
Total number of active requests.

NooBaa_BGWorkers_nodejs_heap_size_total_bytes
Process heap size from Node.js in bytes.

NooBaa_BGWorkers_nodejs_heap_size_used_bytes
Process heap size used from Node.js in bytes.

NooBaa_BGWorkers_nodejs_external_memory_bytes
Node.js external memory size in bytes.

NooBaa_BGWorkers_nodejs_heap_space_size_total_bytes
Process heap space size total from Node.js in bytes.

NooBaa_BGWorkers_nodejs_heap_space_size_used_bytes
Process heap space size used from Node.js in bytes.

NooBaa_BGWorkers_nodejs_heap_space_size_available_bytes
Process heap space size available from Node.js in bytes.

NooBaa_BGWorkers_nodejs_version_info
Node.js version info.

NooBaa_BGWorkers_nodejs_gc_duration_seconds
Garbage collection duration by kind, one of major, minor, incremental or weakcb.

NooBaa_HostedAgents_process_cpu_user_seconds_total
Total user CPU time spent in seconds.

NooBaa_HostedAgents_process_cpu_system_seconds_total
Total system CPU time spent in seconds.

NooBaa_HostedAgents_process_cpu_seconds_total
Total user and system CPU time spent in seconds.

NooBaa_HostedAgents_process_start_time_seconds
Start time of the process since unix epoch in seconds.

NooBaa_HostedAgents_process_resident_memory_bytes
Resident memory size in bytes.

NooBaa_HostedAgents_process_virtual_memory_bytes
Virtual memory size in bytes.

NooBaa_HostedAgents_process_heap_bytes
Process heap size in bytes.

NooBaa_HostedAgents_process_open_fds
Number of open file descriptors.

NooBaa_HostedAgents_process_max_fds
Maximum number of open file descriptors.

NooBaa_HostedAgents_nodejs_eventloop_lag_seconds
Lag of event loop in seconds.

NooBaa_HostedAgents_nodejs_eventloop_lag_min_seconds
The minimum recorded event loop delay.

NooBaa_HostedAgents_nodejs_eventloop_lag_max_seconds
The maximum recorded event loop delay.

NooBaa_HostedAgents_nodejs_eventloop_lag_mean_seconds
The mean of the recorded event loop delays.

NooBaa_HostedAgents_nodejs_eventloop_lag_stddev_seconds
The standard deviation of the recorded event loop delays.

NooBaa_HostedAgents_nodejs_eventloop_lag_p50_seconds
The 50th percentile of the recorded event loop delays.

NooBaa_HostedAgents_nodejs_eventloop_lag_p90_seconds
The 90th percentile of the recorded event loop delays.

NooBaa_HostedAgents_nodejs_eventloop_lag_p99_seconds
The 99th percentile of the recorded event loop delays.

NooBaa_HostedAgents_nodejs_active_resources
Number of active resources that are currently keeping the event loop alive, grouped by async resource type.

NooBaa_HostedAgents_nodejs_active_resources_total
Total number of active resources.

NooBaa_HostedAgents_nodejs_active_handles
Number of active libuv handles grouped by handle type. Every handle type is C++ class name.

NooBaa_HostedAgents_nodejs_active_handles_total
Total number of active handles.

NooBaa_HostedAgents_nodejs_active_requests
Number of active libuv requests grouped by request type. Every request type is C++ class name.
NooBaa_HostedAgents_nodejs_active_requests_total
Total number of active requests.
NooBaa_HostedAgents_nodejs_heap_size_total_bytes
Process heap size from Node.js in bytes.
NooBaa_HostedAgents_nodejs_heap_size_used_bytes
Process heap size used from Node.js in bytes.
NooBaa_HostedAgents_nodejs_external_memory_bytes
Node.js external memory size in bytes.
NooBaa_HostedAgents_nodejs_heap_space_size_total_bytes
Process heap space size total from Node.js in bytes.
NooBaa_HostedAgents_nodejs_heap_space_size_used_bytes
Process heap space size used from Node.js in bytes.
NooBaa_HostedAgents_nodejs_heap_space_size_available_bytes
Process heap space size available from Node.js in bytes.
NooBaa_HostedAgents_nodejs_version_info
Node.js version info.
NooBaa_HostedAgents_nodejs_gc_duration_seconds
Garbage collection duration by kind, one of major, minor, incremental or weakcb.
NooBaa_cloud_types
Cloud Resource Types in the System
NooBaa_projects_capacity_usage
Projects Capacity Usage
NooBaa_accounts_usage_read_count
Accounts Usage Read Count
NooBaa_accounts_usage_write_count
Accounts Usage Write Count
NooBaa_accounts_usage_logical
Accounts Usage Logical
NooBaa_bucket_class_capacity_usage
Bucket Class Capacity Usage
NooBaa_unhealthy_cloud_types
Unhealthy Cloud Resource Types in the System
NooBaa_object_histo
Object Sizes Histogram Across the System
NooBaa_providers_bandwidth_write_size
Providers bandwidth write size
NooBaa_providers_bandwidth_read_size
Providers bandwidth read size
NooBaa_providers_ops_read_num
Providers number of read operations
NooBaa_providers_ops_write_num
Providers number of write operations
NooBaa_providers_physical_size
Providers Physical Stats
NooBaa_providers_logical_size
Providers Logical Stats
NooBaa_system_capacity
System capacity
NooBaa_system_info
System info
NooBaa_num_buckets
Object Buckets
NooBaa_num_namespace_buckets
Namespace Buckets
NooBaa_total_usage
Total Usage
NooBaa_accounts_num
Accounts Number
NooBaa_num_objects
Objects
NooBaa_num_unhealthy_buckets
Unhealthy Buckets
NooBaa_num_unhealthy_namespace_buckets
Unhealthy Namespace Buckets
NooBaa_num_unhealthy_pools
Unhealthy Resource Pools
NooBaa_num_unhealthy_namespace_resources
Unhealthy Namespace Resources
NooBaa_num_pools
Resource Pools
NooBaa_num_namespace_resources
Namespace Resources
NooBaa_num_unhealthy_bucket_claims
Unhealthy Bucket Claims
NooBaa_num_buckets_claims
Object Bucket Claims
NooBaa_num_objects_buckets_claims
Objects On Object Bucket Claims
NooBaa_reduction_ratio
Object Efficiency Ratio
NooBaa_object_savings_logical_size
Object Savings Logical
NooBaa_object_savings_physical_size
Object Savings Physical
NooBaa_rebuild_progress

Rebuild Progress

NooBaa_rebuild_time
Rebuild Time

NooBaa_bucket_status
Bucket Health

NooBaa_namespace_bucket_status
Namespace Bucket Health

NooBaa_bucket_tagging
Bucket Tagging

NooBaa_namespace_bucket_tagging
Namespace Bucket Tagging

NooBaa_bucket_capacity
Bucket Capacity Precent

NooBaa_bucket_size_quota
Bucket Size Quota Precent

NooBaa_bucket_quantity_quota
Bucket Quantity Quota Precent

NooBaa_bucket_object_count
Number of objects currently stored in the bucket.

NooBaa_bucket_max_objects_quota
Maximum number of objects allowed as bucket quota.

NooBaa_bucket_max_bytes_quota
Maximum storage capacity (in bytes) permitted bucket quota.

NooBaa_resource_status
Resource Health

NooBaa_namespace_resource_status
Namespace Resource Health

NooBaa_system_links
System Links

NooBaa_health_status
Health status

NooBaa_odf_health_status
Health status

NooBaa_replication_status
Replication status

NooBaa_replication_last_cycle_writes_size
Number of bytes replicated by replication_id in last replication cycle

NooBaa_replication_last_cycle_writes_num
Number of objects replicated by replication_id in last replication cycle

NooBaa_replication_last_cycle_error_writes_size
Number of error bytes replication_id in last replication cycle

NooBaa_replication_last_cycle_error_writes_num
Number of error objects replication_id in last replication cycle

NooBaa_bucket_used_bytes
Object Bucket Used Bytes

Fusion Data Foundation service install and upgrade issues

Use this troubleshooting information to resolve install and upgrade problems that are related to Fusion Data Foundation service.

- ceph-csi-controller-manager pod in Crashloopbackoff state
- Data Foundation operator upgrade is not initiated
- Data Foundation ocs-client-operator-console-xxx pod is stuck while upgrade

ceph-csi-controller-manager pod in Crashloopbackoff state

Problem statement
After the upgrade of OpenShift® Container Platform 4.16 and Fusion Data Foundation 4.16 to 4.17, the Fusion Data Foundation goes into a degraded state.

Cause
It error occurs because of the **Crashloopbackoff** state of the **ceph-csi-controller-manager** pod.

Resolution
Do the following steps to resolve and validate this error:

1. Run the following example command to remove the limit from CSV:

   ```
   oc patch csv cephcsi-operator.v4.17.1 --type json -p '[ {"op": "remove", "path":
   "/spec/install/spec/deployments/0/spec/template/spec/containers/1/resources/limits"} ]'
   clusterserviceversion.operators.coreos.com/cephcsi-operator.v4.17.1 patched
   ```

2. Run the following command to verify the pods status:

   ```
   oc get pods
   ```

   Example output:

   ```
   NAME                                                              READY   STATUS    RESTARTS      AGE
   ceph-csi-controller-manager-675dc67f7b-f64pb                      2/2     Running   0             3h39m
   ceph-csi-controller-manager-8854c7f5-xdrvx                        1/2     Running   0             6s
   ```

Data Foundation operator upgrade is not initiated

Problem statement

Data Foundation operator upgrade is not initiated even though the service is installed successfully. If auto-upgrade is not enabled for Data Foundation during the upgrade to version 4.17 or later, some components might not upgrade automatically.

Resolution

To ensure all components are upgraded successfully, verify that all `InstallPlans` in the `openshift-storage` namespace are approved. Follow the steps to verify and manually approve any pending `InstallPlans`:

1. Run the following command to retrieve the list of `InstallPlans` in the `openshift-storage` namespace.

   ```
   oc get installplan -n openshift-storage
   ```

2. Check the `APPROVED` column in the output and make sure that all listed `InstallPlans` status set to `true`.
3. If any `InstallPlans` in the `APPROVED` column is set to `false`, then manually approve it by running the following command.

   ```
   for ip in $(oc get installplan --no-headers -n openshift-storage | grep false | awk '{print $1}')
   do
     oc patch installplan $ip -n openshift-storage --type merge --patch '{"spec":{"approved":true}}'
   done
   ```

4. After approving all pending `InstallPlans`, repeat the steps 1 and 2 to confirm that all `InstallPlans` status set to true.

Data Foundation `ocs-client-operator-console-xxx` pod is stuck while upgrade

Problem statement

During the Data Foundation service upgrade from version 4.18.2 to 4.18.3, an error occurred: the `odf-operator` CSV `ocs-operator.v4.18.3` is unavailable due to `ocs-client-operator-console-xxx` pod is stuck in `ContainerCreating` state.

Resolution

Delete the pod to resolve the issue using the following command.

```
oc delete pod <name> -nopenshift-storage
```

Troubleshooting

Administrators can use this troubleshooting information to understand how to troubleshoot and fix their IBM Fusion Data Foundation cluster.

Most troubleshooting tasks focus on either a fix or a workaround. This chapter is divided into sections based on the errors that an administrator may encounter:

- See Downloading log files and diagnostic information using must-gather to understand how to use the must-gather utility in Fusion Data Foundation.
- See Commonly required logs for troubleshooting to understand how to obtain commonly required log files for Fusion Data Foundation.
- See Troubleshooting alerts and errors in Fusion Data Foundation to identify various errors and required actions.

Warning: IBM does not support the use of Ceph toolbox in Fusion Data Foundation clusters as it can cause data loss. If you use Ceph toolbox, IBM Support is only able to provide best-effort support and may not be able to restore all the data in case of any data loss.

- **Downloading log files and diagnostic information using must-gather**
  Use the must-gather image to download logs and diagnostic information.
- **Using ODF CLI command**
  The Red Hat OpenShift Data Foundation command-line interface (CLI) command and its subcommands reduce repetitive tasks and provide a better experience. To download the ODF CLI tool, see customer portal.
- **Commonly required logs for troubleshooting**
  Gather commonly required logs for troubleshooting.
- **Overriding the cluster-wide default node selector for Fusion Data Foundation post deployment**
  When a cluster-wide default node selector is used for Fusion Data Foundation, the pods generated by CSI daemon sets are able to start only on the nodes that match the selector. To be able to use Fusion Data Foundation from nodes which do not match the selector, override the `cluster-wide default node selector`, using the command line interface.
- **Encryption token is deleted or expired**
  Update the token if the encryption token for your key management system gets deleted or expires.
- **Troubleshooting alerts and errors in Fusion Data Foundation**
  Understand the alerts and errors within Fusion Data Foundation to help resolve and recover issues.
- **Checking for Local Storage Operator deployments**
  IBM Fusion Data Foundation clusters with Local Storage Operator are deployed using local storage devices. To find out if your existing cluster with Fusion Data Foundation was deployed using local storage devices, use the following procedure:
- **Removing failed or unwanted Ceph Object Storage devices**
  Failed or unwanted Ceph Object Storage Devices (OSDs) reduce storage cluster reliability and must be removed after verifying Ceph health, using the procedure appropriate for dynamically provisioned or local storage deployments.
- **Troubleshooting and deleting remaining resources during Uninstall**
  Occasionally some of the custom resources managed by an operator may remain in *Terminating* status waiting on the finalizer to complete, although you have performed all the required cleanup tasks. In such an event you need to force the removal of such resources. If you do not do so, the resources remain in the *Terminating* state even after you have performed all the uninstall steps.
- **Troubleshooting CephFS PVC creation in external mode**
  If you have updated the IBM Storage Ceph cluster from a version lower than 4.1.1 to the latest release and is not a freshly deployed cluster, you must manually set the application type for CephFS pool on the IBM Storage Ceph cluster to enable CephFS PVC creation in external mode.
- **Restoring the monitor pods in Fusion Data Foundation**
  Use this information to manually restore the monitor pods Fusion Data Foundation, when necessary.
- **Restoring ceph-monitor quorum in Fusion Data Foundation**
  In some circumstances, the `ceph-mons` might lose quorum. If the `mons` cannot form quorum again, there is a manual procedure to get the quorum going again. The only requirement is that, at least one `mon` must be healthy. Use this information to remove the unhealthy `mons` from quorum and form a quorum again with a single `mon`, then bring the quorum back to the original size.

- **Changing resources for the Fusion Data Foundation components**
  When you install Fusion Data Foundation, it comes with pre-defined resources that the Fusion Data Foundation pods can consume. In some situations with higher I/O load, it might be required to increase these limits.
- **Disabling Multicloud Object Gateway external service after deploying OpenShift Data Foundation**
  When you deploy Fusion Data Foundation, public IPs are created even when OpenShift is installed as a private cluster. However, you can disable the Multicloud Object Gateway (MCG) load balancer usage by using the `disableLoadBalancerService` variable in the storagecluster CRD. This restricts MCG from creating any public resources for private clusters and helps to disable the NooBaa service `EXTERNAL-IP`.
- **Accessing odf-console with the ovs-multitenant plug-in by manually enabling global pod networking**
  If the `odf-console` plug-in fails with a timeout error when using the `ovs-multitenant` SDN plug-in, make the `openshift-storage` project's pod networking global to allow cross-namespace console communication.
- **Annotating encrypted RBD storage classes**
  This section provides you with information on annotating encrypted RBD storage classes.
- **Force deletion of storage in provider clusters**
  Force-delete a storage consumer from the provider cluster when a client cluster was removed without completing the off-boarding process, to release storage space claimed by the deleted client.
- **Configuring CPU and memory resources for FDF-managed pods**
  You can define and adjust resource requests or limits for the pods managed by Fusion Data Foundation (FDF) to ensure optimal scheduling and performance of FDF workloads.
- **Unable to delete incorrect IBM Scale (CNSA) cluster after first-time configuration**
  When incorrect or mock data is entered during first-time IBM Scale (CNSA) configuration with Fusion Data Foundation, a `Cluster` is created and recognized as Storage Area Network (SAN), cannot be deleted, and prevents a valid Scale connection.
- **Filesystem deletion does not remove internal IBM Storage Scale resources**
  Deleting a filesystem custom resource does not automatically remove internal IBM Storage Scale remote cluster definitions.

Downloading log files and diagnostic information using must-gather

Use the must-gather image to download logs and diagnostic information.

Before you begin

Optional: If Fusion Data Foundation is deployed in a disconnected environment, ensure that you mirror the individual `must-gather` image to the mirror registry available from the disconnected environment.

```
oc image mirror registry.redhat.io/odf4/odf-must-gather-rhel9:<odf-version> <local-registry>/odf4/odf-must-gather-rhel9:<odf-version> [--registry-config=<path-to-the-registry-config>] [--insecure=true]
```

`<odf-version>`
: Is the major version number of data foundation. For example, v4.16.

`<local-registry>`
: Is the local image mirror registry available for a disconnected OpenShift Container Platform cluster.

`<path-to-the-registry-config>`
: Is the path to your registry credentials, by default it is ~/.docker/config.json.

`--insecure`
: Add this flag only if the mirror registry is insecure.

For more information, see the Red Hat Knowledgebase solutions:

- How to mirror images between Redhat Openshift registries
- Failed to mirror OpenShift image repository when private registry is insecure

If IBM Fusion Data Foundation is unable to automatically resolve a problem, use the `must-gather` tool to collect log files and diagnostic information so that you or IBM support can review the problem and determine a solution.
Important: When IBM Fusion Data Foundation is deployed in external mode, `must-gather` only collects logs from the Fusion Data Foundation cluster and does not collect debug data and logs from the external IBM Storage Ceph cluster. To collect debug logs from the external IBM Storage Ceph cluster, see Troubleshootingwithin IBM Storage Ceph documentation and contact your IBM Storage Ceph Administrator.

1. Run the **must-gather** command from the client connected to the Fusion Data Foundation cluster, where *<directory-name>* is the name of the directory where you want to write the data to.

   ```
   oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:<odf-version> --dest-dir=<directory-name>
   ```

   Important: For a disconnected environment deployment, replace the image in --image parameter with the mirrored `must-gather` image, where *<local-registry>* is the local image mirror registry available for a disconnected OpenShift Container Platform cluster..

   ```
   oc adm must-gather --image=<local-registry>/odf4/odf-must-gather-rhel9:<odf-version> --dest-dir=<directory-name>
   ```

   This collects the following information in the specified directory:
   - All Fusion Data Foundation cluster related Custom Resources (CRs) with their namespaces.
   - Pod logs of all the Fusion Data Foundation related pods.
   - Output of some standard Ceph commands like *Status*, *Cluster health*, and others.

Command variations

- If one or more master nodes are not in the *Ready* state, use `--node-name` to provide a master node that is *Ready* so that the `must-gather` pod can be safely scheduled.

  ```
  oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:<odf-version> --dest-dir=_<directory-name>_ --node-name=_<node-name>_
  ```

- If you want to gather information from a specific time:
  - To specify a relative time period for logs gathered, such as within 5 seconds or 2 days, add `/usr/bin/gather since=<duration>`:

    ```
    oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:<odf-version> --dest-dir=_<directory-name>_ /usr/bin/gather since=<duration>
    ```

  - To specify a specific time to gather logs after, add `/usr/bin/gather since-time=<rfc3339-timestamp>`:

    ```
    oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:<odf-version> --dest-dir=_<directory-name>_ /usr/bin/gather since-time=<rfc3339-timestamp>
    ```

  Replace the example values in these commands as follows:

  `<node-name>`
  : If one or more master nodes are not in the *Ready* state, use this parameter to provide the name of a master node that is still in the *Ready* state. This avoids scheduling errors by ensuring that the `must-gather` pod is not scheduled on a master node that is not ready.

  `<directory-name>`
  : The directory to store information collected by `must-gather`.

  `<duration>`
  : Specify the period of time to collect information from as a relative duration, for example, `5h` (starting from 5 hours ago).

  `<rfc3339-timestamp>`
  : Specify the period of time to collect information from as an RFC 3339 timestamp, for example, `2020-11-10T04:00:00+00:00` (starting from 4am UTC on 11 Nov 2020).

- **Running must-gather in modular mode**
  The **must-gather** command provides an ability to collect only the required logs. Run the **must-gather** command in modular mode to collect logs for required resources instead of all resources.

Running must-gather in modular mode

The **must-gather** command provides an ability to collect only the required logs. Run the **must-gather** command in modular mode to collect logs for required resources instead of all resources.

The **must-gather** command runs quickly in a modular mode and collects logs for required resources. To collect logs, enter the following command:

```
oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:<odf-version> -- /usr/bin/gather <-arg> --dest-dir=<directory-name>
```

Replace the <-arg> option with one or more of the following options to specify the required resources for collecting logs:

Note: If you run the **must-gather** command without replacing the <-arg> option with the required options or resources, the command collects and displays logs for all resources.

`-o`, `--odf`
: ODF logs (includes Ceph resources, namespaced resources, clusterscoped resources, and Ceph logs)

`-d`, `--dr`
: DR logs

`-n`, `--noobaa`
: Noobaa logs

`-c`, `--ceph`
: Ceph commands and pod logs

`-cl`, `--ceph-logs`
: Ceph daemon, kernel, journal logs, and crash reports

`-ns`, `--namespaced`
: `namespaced` resources

`-cs`, `--clusterscoped`
: `clusterscoped` resources

`-pc`, `--provider`
: `openshift-storage-client` logs from a provider/consumer cluster (includes all the logs under operator namespace, pods, deployments, secrets, configmap, and other resources)

`-h`, `--help`
: Print help message

Using ODF CLI command

The Red Hat OpenShift Data Foundation command-line interface (CLI) command and its subcommands reduce repetitive tasks and provide a better experience. To download the ODF CLI tool, see customer portal.

Subcommands of odf get command

odf get recovery-profile

This command displays the recovery profile value set for the Object Storage Daemon (OSD). If the value is not set that using the **odf set recovery-profile** command, an empty value is displayed by default. When a value is set, the appropriate value is displayed.

Example:

```
 odf get recovery-profile
# high_recovery_ops
```

odf get health

This command checks the health of the Ceph cluster and common configuration issues. This command checks the following configuration for the health of the Ceph cluster:

- At least three monitor (mon) pods are running on different nodes
- Mon quorum and Ceph health details
- At least three OSD pods are running on different nodes
- The running status of all pods
- Placement group status
- At least one MGR pod is running

Example:

```
 odf get health

Info: Checking if at least three mon pods are running on different nodes
rook-ceph-mon-a-7fb76597dc-98pxz        Running openshift-storage       ip-10-0-69-145.us-west-1.compute.internal
rook-ceph-mon-b-885bdc59c-4vvcm Running openshift-storage       ip-10-0-64-239.us-west-1.compute.internal
rook-ceph-mon-c-5f59bb5dbc-8vvlg        Running openshift-storage       ip-10-0-30-197.us-west-1.compute.internal

Info: Checking mon quorum and ceph health details
Info: HEALTH_OK
[...]
```

odf get dr-health

This command fetches the connection status of a cluster from another cluster in mirroring-enabled clusters. This command fetches Ceph block pool with mirroring-enabled information. If the information is not found, this command produces output with relevant logs.

Example:

```
 odf get dr-health

Info: fetching the cephblockpools with mirroring enabled
Info: found "ocs-storagecluster-cephblockpool" cephblockpool with mirroring enabled
Info: running ceph status from peer cluster
Info:   cluster:
    id:     9a2e7e55-40e1-4a79-9bfa-c3e4750c6b0f
    health: HEALTH_OK

[...]
```

odf get dr-prereq

This command checks and fetches the status of all the prerequisites to enable disaster recovery on a pair of clusters. The command takes the peer cluster name as an argument and uses it to compare the current cluster configuration with the peer cluster configuration. Based on the comparison results, it displays the status of the prerequisites.

Example:

```
 odf get dr-prereq peer-cluster-1

Info: Submariner is installed.
Info: Globalnet is required.
Info: Globalnet is enabled.


odf get mon-endpoints
Displays the mon endpoints
$ odf get dr-prereq peer-cluster-1

Info: Submariner is installed.
Info: Globalnet is required.
Info: Globalnet is enabled.
```

Subcommands of odf operator command

odf operator rook set

This command sets the provided property value in the Rook Ceph operator configuration.

Example:

```
 odf operator rook set ROOK_LOG_LEVEL DEBUG

configmap/rook-ceph-operator-config patched
```

The value of the *ROOK_LOG_LEVEL* can be *DEBUG*, *INFO*, or *WARNING*.

odf operator rook restart

This command restarts the Rook Ceph operator.

Example:

```
odf operator rook restart

deployment.apps/rook-ceph-operator restarted
```

odf restore mon-quorum

This command restores the mon quorum when the majority of mons are not in quorum and the cluster is down. When the majority of mons are lost permanently, restoring the quorum to a remaining good mon to bring the Ceph cluster up again is necessary.

Example:

```
odf restore mon-quorum c
```

odf restore deleted <crd>

This command restores the deleted Rook Custom Resource (CR) when data is left for the components, Ceph clusters, Ceph file systems, and Ceph block pools. Generally, when Rook CR is deleted and there is leftover data, the Rook operator does not delete the CR to ensure data is not lost, and the operator does not remove the finalizer on the CR. As a result, the CR is stuck in the deleting state, and cluster health is not ensured. Upgrades are blocked too. This command helps repair the CR without the cluster downtime.

Important: A warning message seeking confirmation to restore appears. After confirming, you need to enter continue to start the operator and expand to the full mon-quorum again.
Example:

```
odf restore deleted cephclusters

Info: Detecting which resources to restore for crd "cephclusters"

Info: Restoring CR my-cluster
Warning: The resource my-cluster was found deleted. Do you want to restore it? yes | no
[...]
```

Commonly required logs for troubleshooting

Gather commonly required logs for troubleshooting.

Some of the commonly used logs for troubleshooting Fusion Data Foundation are listed, along with the commands to generate them.

- Generating logs for a specific pod:

  ```
  oc logs <pod-name> -n <namespace>
  ```

- Generating logs for Ceph or Fusion Data Foundation cluster:

  ```
  oc logs rook-ceph-operator-<ID> -n openshift-storage
  ```

  Important: Currently, the `rook-ceph-operator` logs do not provide any information about the failure and this acts as a limitation in troubleshooting issues, see Enabling debug logs for rook-ceph-operator and Disabling debug logs for rook-ceph-operator.
- Generating logs for plugin pods like cephfs or rbd to detect any problem in the PVC mount of the app-pod:

  ```
  oc logs openshift-storage.cephfs.csi.ceph.com-ctrlplugin-<ID> -n openshift-storage -c openshift-storage.cephfs.csi.ceph.com-ctrlplugin

  oc logs openshift-storage.rbd.csi.ceph.com-ctrlplugin-<ID> -n openshift-storage -c openshift-storage.rbd.csi.ceph.com-ctrlplugin
  ```

  - To generate logs for all the containers in the CSI pod:

    ```
    oc logs openshift-storage.cephfs.csi.ceph.com-ctrlplugin-<ID> -n openshift-storage --all-containers

    oc logs openshift-storage.rbd.csi.ceph.com-ctrlplugin-<ID> -n openshift-storage --all-containers
    ```

- Generating logs for cephfs or rbd nodeplugin pods to detect problems if PVC is not in *BOUND* state:

  ```
  oc logs openshift-storage.cephfs.csi.ceph.com-nodeplugin-<ID> -n openshift-storage -c openshift-storage.cephfs.csi.ceph.com-ctrlplugin

  oc logs openshift-storage.rbd.csi.ceph.com-nodeplugin-<ID> -n openshift-storage -c openshift-storage.rbd.csi.ceph.com-ctrlplugin
  ```

  - To generate logs for all the containers in the CSI pod:

    ```
    oc logs openshift-storage.cephfs.csi.ceph.com-nodeplugin-<ID> -n openshift-storage --all-containers

    oc logs openshift-storage.rbd.csi.ceph.com-nodeplugin-<ID> -n openshift-storage --all-containers
    ```

- Generating Fusion Data Foundation logs using cluster-info command:

```
oc cluster-info dump -n openshift-storage --output-directory=<directory-name>
```

- When using Local Storage Operator, generating logs can be done using cluster-info command:

  ```
  oc cluster-info dump -n openshift-local-storage --output-directory=<directory-name>
  ```

- Check the Fusion Data Foundation operator logs and events.
  - To check the operator logs :

    ```
    oc logs <ocs-operator> -n openshift-storage
    ```

    ```
    <ocs-operator>
    ```

    ```
    oc get pods -n openshift-storage | grep -i "ocs-operator" | awk '{print $1}'
    ```

  - To check the operator events :

    ```
    oc get events --sort-by=metadata.creationTimestamp -n openshift-storage
    ```

- Get the Fusion Data Foundation operator version and channel.

  ```
  oc get csv -n openshift-storage
  ```

  Example output :

  ```
  NAME                              DISPLAY                       VERSION   REPLACES   PHASE
  mcg-operator.v4.12.0              NooBaa Operator               4.12.0               Succeeded
  ocs-operator.v4.12.0              OpenShift Container Storage   4.12.0               Succeeded
  odf-csi-addons-operator.v4.12.0   CSI Addons                    4.12.0               Succeeded
  odf-operator.v4.12.0              Fusion Data Foundation     4.12.0              Succeeded
  ```

  ```
  oc get subs -n openshift-storage
  ```

  Example output:

  ```
  NAME                                                              PACKAGE                   SOURCE             CHANNEL
  mcg-operator-stable-4.12-redhat-operators-openshift-marketplace   mcg-operator              redhat-operators   stable-4.12
  ocs-operator-stable-4.12-redhat-operators-openshift-marketplace   ocs-operator              redhat-operators   stable-4.12
  odf-csi-addons-operator                                           odf-csi-addons-operator   redhat-operators   stable-4.12
  odf-operator                                                      odf-operator              redhat-operators   stable-4.12
  ```

- Confirm that the installplan is created.

  ```
  oc get installplan -n openshift-storage
  ```

- Verify the image of the components post updating Fusion Data Foundation.
  - Check the node on which the pod of the component you want to verify the image is running.

    ```
    oc get pods -o wide | grep <component-name>
    ```

    For Example :

    ```
    oc get pods -o wide | grep rook-ceph-operator
    ```

    Example output, where `dell-r440-12.gsslab.pnq2.redhat.com` is the *node-name*:

    ```
    rook-ceph-operator-566cc677fd-bjqnb 1/1 Running 20 4h6m 10.128.2.5 rook-ceph-operator-566cc677fd-bjqnb 1/1 Running 20 4h6m 10.128.2.5 dell-r440-12.gsslab.pnq2.redhat.com <none> <none>
    ```

    ```
    <none> <none>
    ```

  - Check the image ID, where *<node-name>* is the name of the node on which the pod of the component you want to verify the image is running.

    ```
    oc debug node/<node name>
    ```

    ```
    chroot /host
    ```

    ```
    crictl images | grep <component>
    ```

    For Example :

    ```
    crictl images | grep rook-ceph
    ```

    Take a note of the `IMAGEID` and map it to the *Digest* ID on the Rook Ceph Operator page.

For more information, see Using must-gather.

- **Adjusting verbosity level of logs**
  Adjusting verbosity levels of CSI logs controls the storage space consumed by debugging output; set `CSI_LOG_LEVEL` and `CSI_SIDECAR_LOG_LEVEL` parameters in the container YAML to values between 0 and 5.

Adjusting verbosity level of logs

Adjusting verbosity levels of CSI logs controls the storage space consumed by debugging output; set `CSI_LOG_LEVEL` and `CSI_SIDECAR_LOG_LEVEL` parameters in the container YAML to values between 0 and 5.

The amount of space consumed by debugging logs can become a significant issue. Fusion Data Foundation offers a method to adjust, and therefore control, the amount of storage to be consumed by debugging logs.

To adjust the verbosity levels of debugging logs, you can tune the log levels of the containers responsible for CSI operations. In the container's yaml file, adjust the following parameters to set the logging levels:

- `CSI_LOG_LEVEL` - defaults to `5`
- `CSI_SIDECAR_LOG_LEVEL` - defaults to `1`

The supported values are `0` through `5`. Use `0` for general useful logs, and `5` for trace level verbosity.

Overriding the cluster-wide default node selector for Fusion Data Foundation post deployment

When a cluster-wide default node selector is used for Fusion Data Foundation, the pods generated by CSI daemon sets are able to start only on the nodes that match the selector. To be able to use Fusion Data Foundation from nodes which do not match the selector, override the `cluster-wide default node selector`, using the command line interface.

Procedure

1. Specify a blank node selector for the openshift-storage namespace.

   ```
   oc annotate namespace openshift-storage openshift.io/node-selector=
   ```

2. Delete the original pods generated by the DaemonSets.

   ```
   oc delete pod -l app=openshift-storage.cephfs.csi.ceph.com-ctrlplugin -n openshift-storage
   oc delete pod -l app=openshift-storage.rbd.csi.ceph.com-ctrlplugin -n openshift-storage
   ```

Encryption token is deleted or expired

Update the token if the encryption token for your key management system gets deleted or expires.

Before you begin

Ensure that you have a new token with the same policy as the deleted or expired token.

About this task

Use this procedure to update the token if the encryption token for your key management system gets deleted or expires.

Procedure

1. Log in to OpenShift Container Platform Web Console.
2. Go to Workloads > Secrets.
3. Update the ocs-kms-token used for cluster wide encryption.
   a. Set the Project to `openshift-storage`.
   b. Go to ocs-kms-token > Actions > Edit Secret.
   c. Drag and drop or upload your encryption token file in the Value field.
      The token can either be a file or text that can be copied and pasted.
   d. Click Save.
4. Update the ceph-csi-kms-token for a given project or namespace with encrypted persistent volumes.
   a. Select the required Project.
   b. Go to ceph-csi-kms-token > Actions > Edit Secret.
   c. Drag and drop or upload your encryption token file in the Value field.
      The token can either be a file or text that can be copied and pasted.
   d. Click Save.
      Note: The token can be deleted only after all the encrypted PVCs using the `ceph-csi-kms-token` have been deleted.

Troubleshooting alerts and errors in Fusion Data Foundation

Understand the alerts and errors within Fusion Data Foundation to help resolve and recover issues.

- **Resolving alerts and errors**
  IBM Fusion Data Foundation can detect and automatically resolve a number of common failure scenarios. However, some problems require administrator intervention. Use this information to resolve alerts and errors.
- **Resolving cluster health issues**
  There is a finite set of possible health messages that a IBM Storage Ceph cluster can raise that show in the Fusion Data Foundation user interface. These are defined as health checks which have unique identifiers. See the health code for more information and troubleshooting.
- **Resolving cluster alerts**
  There is a finite set of possible health alerts that a IBM Storage Ceph cluster can raise that show in the Fusion Data Foundation user interface. These are defined as

health alerts which have unique identifiers. The identifier is a terse pseudo-human-readable string that is intended to enable tools to make sense of health checks, and present them in a way that reflects their meaning.

- **Resolving NooBaa Bucket Error State**
  Use this information to resolve a NooBaa Bucket Error State.
- **Resolving NooBaa Bucket Exceeding Quota State**
  Use this information to resolve a `A NooBaa Bucket Is In Exceeding Quota State` error.
- **Resolving NooBaa Bucket Capacity or Quota State**
  Use this information to resolve a NooBaa Bucket Capacity or Quota State.
- **Recovering pods**
  Use this information to recover a pod.
- **Recovering from EBS volume detach**
  When an OSD or MON elastic block storage (EBS) volume where the OSD disk resides is detached from the worker Amazon EC2 instance, the volume gets reattached automatically within one or two minutes. However, the OSD pod gets into a **`CrashLoopBackOff`** state.
- **Enabling debug logs for rook-ceph-operator**
  Enable the debug logs for the rook-ceph-operator to obtain information about failures that help in troubleshooting issues.
- **Disabling debug logs for rook-ceph-operator**
  Disable the debug logs for the rook-ceph-operator.
- **Resolving low Ceph monitor count alert**
  The **CephMonLowNumber** alert is displayed in the notification panel or Alert Center of the OpenShift Web Console to indicate the low number of Ceph monitor count when your internal mode deployment has five or more nodes, racks, or rooms, and when there are five or more number of failure domains in the deployment. You can increase the Ceph monitor count to improve the availability of cluster.
- **Troubleshooting unhealthy blocklisted nodes**
  If an RBD client on a Kubernetes node is blocklisted by Ceph, causing **`CreateContainerError`** pod states and potentially read-only PVCs, taint the node to evict pods or reboot the node to clear the blocklist.

Resolving alerts and errors

IBM Fusion Data Foundation can detect and automatically resolve a number of common failure scenarios. However, some problems require administrator intervention. Use this information to resolve alerts and errors.

To know the errors currently firing, check any of the following locations:

- Observe > Alerting > Firing
- Home > Overview > Cluser
- Storage > Data Foundation > Storage System click the storage system link in the pop up and then go to Overview > Block and File
- Storage > Data Foundation > Storage System click the storage system link in the pop up and then go to Overview > Object

Use the following information to search and understand the displayed error(s), and to understand its severity and resolution.
Copy the error displayed and search it in the following section to know its severity and resolution:

`CephMonVersionMismatch`

Message: `There are multiple versions of storage services running.`
Description: `There are {{ $value }} different versions of Ceph Mon components running.`

Severity: Warning

Resolution: Fix

Procedure: Inspect the user interface and log, and verify if an update is in progress.

- If an update in progress, this alert is temporary.
- If an update is not in progress, restart the upgrade process.

`CephOSDVersionMismatch`

Message: `There are multiple versions of storage services running.`
Description: `There are {{ $value }} different versions of Ceph OSD components running.`

Severity: Warning

Resolution: Fix

Procedure: Inspect the user interface and log, and verify if an update is in progress.

- If an update in progress, this alert is temporary.
- If an update is not in progress, restart the upgrade process.

`CephClusterCriticallyFull`

Message: `Storage cluster is critically full and needs immediate expansion`
Description: `Storage cluster utilization has crossed 85%.`

Severity: Critical

Resolution: Fix

Procedure: Remove unnecessary data or expand the cluster.

`CephClusterNearFull`

Message: `Storage cluster is nearing full. Expansion is required.`
Description: `Storage cluster utilization has crossed 75%.`

Severity: Warning

Resolution: Fix

Procedure: Remove unnecessary data or expand the cluster.

NooBaaBucketErrorState

Message: A NooBaa Bucket Is In Error State
Description: A NooBaa bucket {{ $labels.bucket_name }} is in error state for more than 6m

Severity: Warning

Resolution: Workaround

Procedure: Resolving NooBaa Bucket Error State

NooBaaNamespaceResourceErrorState

Message: A NooBaa Namespace Resource Is In Error State
Description: A NooBaa namespace resource {{ $labels.namespace_resource_name }} is in error state for more than 5m

Severity: Warning

Resolution: Fix

Procedure: Resolving NooBaa Bucket Error State

NooBaaBucketExceedingQuotaState

Message: A NooBaa Bucket Is In Exceeding Quota State
Description: A NooBaa bucket {{ $labels.bucket_name }} is exceeding its quota - {{ printf "%0.0f" $value }}% used message: A NooBaa Bucket Is In Exceeding Quota State

Severity: Warning

Resolution: Fix

Procedure: Resolving NooBaa Bucket Exceeding Quota State

NooBaaBucketLowCapacityState

Message: A NooBaa Bucket Is In Low Capacity State
Description: A NooBaa bucket {{ $labels.bucket_name }} is using {{ printf "%0.0f" $value }}% of its capacity

Severity: Warning

Resolution: Fix

Procedure: Resolving NooBaa Bucket Capacity or Quota State

NooBaaBucketNoCapacityState

Message: A NooBaa Bucket Is In No Capacity State
Description: A NooBaa bucket {{ $labels.bucket_name }} is using all of its capacity

Severity: Warning

Resolution: Fix

Procedure: Resolving NooBaa Bucket Capacity or Quota State

NooBaaBucketReachingQuotaState

Message: A NooBaa Bucket Is In Reaching Quota State
Description: A NooBaa bucket {{ $labels.bucket_name }} is using {{ printf "%0.0f" $value }}% of its quota

Severity: Warning

Resolution: Fix

Procedure: Resolving NooBaa Bucket Capacity or Quota State

NooBaaResourceErrorState

Message: A NooBaa Resource Is In Error State
Description: A NooBaa resource {{ $labels.resource_name }} is in error state for more than 6m

Severity: Warning

Resolution: Workaround

Procedure: Resolving NooBaa Bucket Error State

NooBaaSystemCapacityWarning100

Message: A NooBaa System Approached Its Capacity
Description: A NooBaa system approached its capacity, usage is at 100%

Severity: Warning

Resolution: Fix

Procedure: Resolving NooBaa Bucket Capacity or Quota State

NooBaaSystemCapacityWarning85

Message: A NooBaa System Is Approaching Its Capacity
Description: A NooBaa system is approaching its capacity, usage is more than 85%

Severity: Warning

Resolution: Fix

Procedure: Resolving NooBaa Bucket Capacity or Quota State

CephMdsMissingReplicas

Message: `Insufficient replicas for storage metadata service.`
Description: `Minimum required replicas for storage metadata service not available. Might affect the working of storage cluster.`

Severity: Warning

Resolution: Contact IBM Support.

Procedure:

1. Check for alerts and operator status.
2. If the issue cannot be identified, contact IBM Support.

CephMgrIsAbsent

Message: `Storage metrics collector service not available anymore.`
Description: `Ceph Manager has disappeared from Prometheus target discovery.`

Severity: Critical

Resolution: Contact IBM Support.

Procedure:

1. Inspect the user interface and log, and verify if an update is in progress.
   - If an update in progress, this alert is temporary.
   - If an update is not in progress, restart the upgrade process.
2. After the upgrade is complete, check for alerts and operator status.
3. If the issue persistents or cannot be identified, contact IBM Support.

CephNodeDown

Message: `Storage node {{ $labels.node }} went down`
Description: `Storage node {{ $labels.node }} went down. Please check the node immediately.`

Severity: Critical

Resolution: Contact IBM Support.

Procedure:

1. Check which node stopped functioning and its cause.
2. Take appropriate actions to recover the node. If node cannot be recovered:
   - See Replacing nodes.
   - Contact IBM Support.

CephClusterErrorState

Message: `Storage cluster is in error state`
Description: `Storage cluster is in error state for more than 10m.`

Severity: Critical

Resolution: Contact IBM Support.

Procedure:

1. Check for alerts and operator status.
2. If the issue cannot be identified, download log files and diagnostic information using must-gather.
3. Open a support ticket with IBM Support. Be sure to attach an output of the must-gather.

CephClusterWarningState

Message: `Storage cluster is in degraded state`
Description: `Storage cluster is in warning state for more than 10m.`

Severity: Warning

Resolution: Contact IBM Support.

Procedure:

1. Check for alerts and operator status.
2. If the issue cannot be identified, download log files and diagnostic information using must-gather.
3. Open a support ticket with IBM Support. Be sure to attach an output of the must-gather.

CephDataRecoveryTakingTooLong

Message: `Data recovery is slow`
Description: `Data recovery has been active for too long.`

Severity: Warning

Resolution: Contact IBM Support.

CephOSDDiskNotResponding

Message: `Disk not responding`
Description: `Disk device {{ $labels.device }} not responding, on host {{ $labels.host }}.`

Severity: Critical

Resolution:

Resolution: Contact IBM Support.

`CephOSDDiskUnavailable`

Message: `Disk not accessible`
Description: `Disk device {{ $labels.device }} not accessible on host {{ $labels.host }}.`

Severity: Critical

Resolution:

Resolution: Contact IBM Support.

`CephPGRepairTakingTooLong`

Message: `Self heal problems detected`
Description: `Self heal operations taking too long.`

Severity: Warning

Resolution: Contact IBM Support.

`CephMonHighNumberOfLeaderChanges`

Message: `Storage Cluster has seen many leader changes recently.`
Description: `'Ceph Monitor "{{ $labels.job }}": instance {{ $labels.instance }} has seen {{ $value printf "%.2f" }} leader changes per minute recently.'`

Severity: Warning

Resolution: Contact IBM Support.

`CephMonQuorumAtRisk`

Message: `Storage quorum at risk`
Description: `Storage cluster quorum is low.`

Severity: Critical

Resolution: Contact IBM Support.

`ClusterObjectStoreState`

Message: `Cluster Object Store is in unhealthy state. Please check Ceph cluster health`
Description: `Cluster Object Store is in unhealthy state for more than 15s. Please check Ceph cluster health`

Severity: Critical

Resolution: Contact IBM Support.

Procedure:

- Check the **`CephObjectStore`** CR instance.
- Contact IBM Support.

`CephOSDFlapping`

Message: `Storage daemon osd.x has restarted 5 times in the last 5 minutes. Please check the pod events or Ceph status to find out the cause`
Description: `Storage OSD restarts more than 5 times in 5 minutes`

Severity: Critical

Resolution: Contact IBM Support.

`OdfPoolMirroringImageHealth`

Message: `Mirroring image(s) (PV) in the pool <pool-name> are in Warning state for more than a 1m. Mirroring might not work as expected.`
Description: Disaster recovery is failing for one or a few applications.

Severity: Warning

Resolution: Contact IBM Support.

`OdfMirrorDaemonStatus`

Message: `Mirror daemon is unhealthy`
Description: Disaster recovery is failing for the entire cluster. Mirror daemon is in unhealthy status for more than 1m. Mirroring on this cluster is not working as expected.

Severity: Critical

Resolution: Contact IBM Support.

`NooBaaDatabaseReachingCapacity`

Message: `NooBaa database on pod noobaa-db-pg-cluster-1 or noobaa-db-pg-cluster-2 is using 80% of its PVC capacity.`
Description: One of the NooBaa database instance is low on disk space and is using 80% of its PVC requested size.

Severity: Warning

Resolution: Expand the PV according to the steps provided in the knowledgebase article, Expanding the db-noobaa-db-pg PVC or contact IBM Support.

`NooBaaDatabaseStorageFull`

Message: `NooBaa database on pod noobaa-db-pg-cluster-1 or noobaa-db-pg-cluster-2 is using over 90% of its PVC capacity.`
Description: One of the NooBaa database instance is low on disk space and is using over 90% of its PVC requested size.

Severity: Critical

Resolution: Expand the PV according to the steps provided in the knowledgebase article, Expanding the db-noobaa-db-pg PVC immediately or contact IBM Support.

Resolving cluster health issues

There is a finite set of possible health messages that a IBM Storage Ceph cluster can raise that show in the Fusion Data Foundation user interface. These are defined as health checks which have unique identifiers. See the health code for more information and troubleshooting.

The identifier is a terse pseudo-human-readable string that is intended to enable tools to make sense of health checks, and present them in a way that reflects their meaning.

- **MON_DISK_LOW**
  One or more Ceph Monitors are low on disk space.

MON_DISK_LOW

One or more Ceph Monitors are low on disk space.

This alert triggers if the available space on the file system storing the monitor database as a percentage, drops below `mon_data_avail_warn` (default: 15%). This may indicate that some other process or user on the system is filling up the same file system used by the monitor. It may also indicate that the monitor's database is large.

Note: The paths to the file system differ depending on the deployment of your mons. You can find the path to where the mon is deployed in storagecluster.yaml. Example paths:

- Mon deployed over PVC path: /var/lib/ceph/mon
- Mon deployed over hostpath: /var/lib/rook/mon

To clear up space, view the high usage files in the file system and choose which to delete. To view the files, run:

```
du -a <path-in-the-mon-node> |sort -n -r |head -n10
```

Replace *<path-in-the-mon-node>* with the path to the file system where mons are deployed.

Resolving cluster alerts

There is a finite set of possible health alerts that a IBM Storage Ceph cluster can raise that show in the Fusion Data Foundation user interface. These are defined as health alerts which have unique identifiers. The identifier is a terse pseudo-human-readable string that is intended to enable tools to make sense of health checks, and present them in a way that reflects their meaning.

See the following sections for more information and troubleshooting.

- **CephClusterCriticallyFull**
  Storage cluster utilization has crossed 80% and becomes read-only at 85%. Your Ceph cluster becomes read-only once utilization crosses 85%. Free up some space or expand the storage cluster immediately. It is common to see alerts related to Object Storage Device (OSD) devices full or near full prior to this alert.
- **CephClusterErrorState**
  This alert reflects that the storage cluster is in *ERROR* state for an unacceptable amount of time and thispts the storage availability. Check for other alerts that would have triggered prior to this one and troubleshoot those alerts first.
- **CephClusterNearFull**
  Storage cluster utilization has crossed 75% and becomes read-only at 85%. Free up some space or expand the storage cluster.
- **CephClusterReadOnly**
  Storage cluster utilization has crossed 85% and is now read-only. Free up some space or expand the storage cluster immediately.
- **CephClusterWarningState**
  This alert reflects that the storage cluster has been in a warning state for an unacceptable amount of time. While the storage operations continue to function in this state, it is recommended to fix the errors so that the cluster does not get into an error statepting operations. Check for other alerts that might have triggered prior to this one and troubleshoot those alerts first.
- **CephDataRecoveryTakingTooLong**
  Data recovery is slow. Check whether all the Object Storage Devices (OSDs) are up and running.
- **CephFSStaleSubvolume**
  A subvolume in a Ceph filesystem becomes stale when it loses connection to its associated PersistentVolume. This results in orphaned data that consumes storage space but remains inaccessible to the user. This alert is triggered when stale subvolumes exist in a Ceph filesystem.
- **CephMdsCpuUsageHigh**
  The storage metadata service (MDS) serves as file system metadata. The MDS is crucial for any file creation, rename, deletion, and update operations. By default, MDS is allocated for two or three CPUs, which does not cause issues usually when not too many metadata operations are on going. When the metadata operation load increases enough to trigger this alert, it means that the default CPU allocation is unable to cope with the load. In such a scenario, increase the CPU allocation or run multiple active MDS servers.
- **CephMdsMissingReplicas**
  Minimum required replicas for the storage metadata service (MDS) are not available. MDS is responsible for filing metadata. Degradation of the MDS service can affect how the storage cluster works (related to the CephFS storage class) and should be fixed as soon as possible.
- **CephMgrIsAbsent**
  Not having a Ceph manager runningpts the monitoring of the cluster. Persistent Volume Claim (PVC) creation and deletion requests should be resolved as soon as possible.
- **CephMgrIsMissingReplicas**
  To resolve this alert, you need to determine the cause of the disappearance of the Ceph manager and restart if necessary.

- **CephMonHighNumberOfLeaderChanges**
  In a Ceph cluster there is a redundant set of monitor pods that store critical information about the storage cluster. Monitor pods synchronize periodically to obtain information about the storage cluster. The first monitor pod to get the most updated information becomes the leader, and the other monitor pods start their synchronization process after asking the leader. A problem in network connection or another kind of problem in one or more monitor pods produces an unusual change of the leader. This situation can negatively affect the storage cluster performance.
- **CephMonQuorumAtRisk**
  Multiple MONs work together to provide redundancy. Each of the MONs keeps a copy of the metadata. The cluster is deployed with 3 MONs, and requires 2 or more MONs to be up and running for quorum and for the storage operations to run. If quorum is lost, access to data is at risk.
- **CephMonQuorumLost**
  In a Ceph cluster there is a redundant set of monitor pods that store critical information about the storage cluster. Monitor pods synchronize periodically to obtain information about the storage cluster. The first monitor pod to get the most updated information becomes the leader, and the other monitor pods start their synchronization process after asking the leader. A problem in network connection or another kind of problem in one or more monitor pods produces an unusual change of the leader. This situation can negatively affect the storage cluster performance.
- **CephMonVersionMismatch**
  Typically this alert triggers during an upgrade that is taking a long time.
- **CephNodeDown**
  A node running Ceph pods is down. While storage operations continue to function as Ceph is designed to deal with a node failure, it is recommended to resolve the issue to minimize the risk of another node going down and affecting storage functions.
- **CephOSDCriticallyFull**
  One of the Object Storage Devices (OSDs) is critically full. Expand the cluster immediately.
- **CephOSDDiskNotResponding**
  A disk device is not responding. Check whether all the Object Storage Devices (OSDs) are up and running.
- **CephOSDDiskUnavailable**
  A disk device is not accessible on one of the hosts and its corresponding Object Storage Device (OSD) is marked out by the Ceph cluster. This alert is raised when a Ceph node fails to recover within 10 minutes.
- **CephOSDFlapping**
  A storage daemon has restarted 5 times in the last 5 minutes. Check the pod events or Ceph status to find out the cause.
- **CephOSDNearFull**
  Utilization of back-end storage device Object Storage Device (OSD) has crossed 75% on a host.
- **CephOSDSlowOps**
  An Object Storage Device (OSD) with slow requests is every OSD that is not able to service the I/O operations per second (IOPS) in the queue within the time defined by the osd_op_complaint_time parameter. By default, this parameter is set to 30 seconds.
- **CephOSDVersionMismatch**
  Typically this alert triggers during an upgrade that is taking a long time.
- **CephPGRepairTakingTooLong**
  Self-healing operations are taking too long.
- **CephPoolFull**
  One or more storage pools have reached 85% capacity on storage cluster. Write operations to affected pools are blocked.
- **CephPoolNearFull**
  One or more storage pools are approaching capacity limits of 75% on storage cluster. Consider adding more storage capacity.
- **CephPoolQuotaBytesCriticallyExhausted**
  One or more pools has reached, or is very close to reaching, its quota. The threshold to trigger this error condition is controlled by the `mon_pool_quota_crit_threshold` configuration option.
- **CephPoolQuotaBytesNearExhaustion**
  One or more pools is approaching a configured fullness threshold. One threshold that can trigger this warning condition is the `mon_pool_quota_warn_threshold` configuration option.
- **CephXattrSetLatency**
  Ceph Metadata Server (MDS) is experiencing high latency when setting extended attributes (xattr) on files. The alert fires when the average setxattr operation latency exceeds 30 milliseconds over a 5-minute period.
- **OSDCPULoadHigh**
  OSD is a critical component in Ceph storage, responsible for managing data placement and recovery. High CPU usage in the OSD container suggests increased processing demands, potentially leading to degraded storage performance.
- **PersistentVolumeUsageCritical**
  A Persistent Volume Claim (PVC) is nearing its full capacity and may lead to data loss if not attended to in a timely manner.
- **PersistentVolumeUsageNearFull**
  A Persistent Volume Claim (PVC) is nearing its full capacity and may lead to data loss if not attended to in a timely manner.
- **NooBaaSystemCapacityWarning85**
  The NooBaa system capacity is calculated by aggregating the total storage capacity of all backingstore resources. Namespacestore resources are not included in this calculation. This alert is triggered when the total used space across the backingstore resources exceeds 85% of the total available capacity.
- **NooBaaSystemCapacityWarning95**
  The NooBaa system capacity is calculated by aggregating the total storage capacity of all backingstore resources. Namespacestore resources are not included in this calculation. This alert is triggered when the total used space across the backingstore resources exceeds 95% of the total available capacity.
- **NooBaaSystemCapacityWarning100**
  The NooBaa system capacity is calculated by aggregating the total storage capacity of all backingstore resources. Namespacestore resources are not included in this calculation. This alert is triggered when the total used space across the backingstore resources reaches 100% of the available capacity.

CephClusterCriticallyFull

Storage cluster utilization has crossed 80% and becomes read-only at 85%. Your Ceph cluster becomes read-only once utilization crosses 85%. Free up some space or expand the storage cluster immediately. It is common to see alerts related to Object Storage Device (OSD) devices full or near full prior to this alert.

Impact: High

Diagnosis

Scaling storage
: Depending on the type of cluster, you need to add storage devices, nodes, or both.

For more information, see Scaling out storage capacity.

Mitigation

Deleting information
If it is not possible to scale up the cluster, you need to delete information in order to free up some space.

CephClusterErrorState

This alert reflects that the storage cluster is in *ERROR* state for an unacceptable amount of time and thispts the storage availability. Check for other alerts that would have triggered prior to this one and troubleshoot those alerts first.

Impact: Critical

Diagnosis

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep rook-ceph**
2. Set `MYPOD` as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
   Examine the output for a rook-ceph that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.
pod status: NOT pending, but NOT running
Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important:

- If a node was assigned, check the kubelet on the node.
- If the basic health of the running pods, node affinity and resource availability on the nodes are verified, run the Ceph tools to get the status of the storage components.

Mitigation

(Optional) Debugging log information
Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephClusterNearFull

Storage cluster utilization has crossed 75% and becomes read-only at 85%. Free up some space or expand the storage cluster.

Impact: Critical

Diagnosis

Scaling storage
Depending on the type of cluster, you need to add storage devices, nodes, or both.
For more information, see Scaling out storage capacity.

Mitigation

Deleting information
If it is not possible to scale up the cluster, you need to delete information in order to free up some space.

CephClusterReadOnly

Storage cluster utilization has crossed 85% and is now read-only. Free up some space or expand the storage cluster immediately.

Impact: Critical

Diagnosis

Scaling storage
: Depending on the type of cluster, you need to add storage devices, nodes, or both.
For more information, see Scaling out storage capacity.

Mitigation

Deleting information
: If it is not possible to scale up the cluster, you need to delete information in order to free up some space.

CephClusterWarningState

This alert reflects that the storage cluster has been in a warning state for an unacceptable amount of time. While the storage operations continue to function in this state, it is recommended to fix the errors so that the cluster does not get into an error statepting operations. Check for other alerts that might have triggered prior to this one and troubleshoot those alerts first.

Impact: High

Diagnosis

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep {ceph-component}**
2. Set `MYPOD` as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
   Examine the output for a {ceph-component} that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
: Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.

pod status: NOT pending, but NOT running
: Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important: If a node was assigned, check the kubelet on the node.

Mitigation

(Optional) Debugging log information
: Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephDataRecoveryTakingTooLong

Data recovery is slow. Check whether all the Object Storage Devices (OSDs) are up and running.

Impact: High

Diagnosis

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep rook-ceph-osd**
2. Set `MYPOD` as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
   Examine the output for a {ceph-component} that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
: Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.

pod status: NOT pending, but NOT running
: Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important: If a node was assigned, check the kubelet on the node.

Mitigation

(Optional) Debugging log information

Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephFSStaleSubvolume

A subvolume in a Ceph filesystem becomes stale when it loses connection to its associated PersistentVolume. This results in orphaned data that consumes storage space but remains inaccessible to the user. This alert is triggered when stale subvolumes exist in a Ceph filesystem.

Impact: High

Diagnosis

To identify stale subvolumes in the filesystem, you need to use the ODF CLI tool.

1. Install ODF CLI
   If not already installed, follow the installation instructions at: Using the ODF CLI command.
2. List all subvolumes including those marked as stale:

   ```
   odf subvolume ls
   ```
3. List only the stale subvolumes:

   ```
   odf subvolume ls --stale
   ```

Mitigation

If the stale subvolumes are no longer needed and not currently in use, you can safely delete them to reclaim storage space:

```
odf subvolume delete <filesystem> <subvolume> <subvolumegroup>
```

The `subvolume`, `subvolumegroup`, and `filesystem` details can be retrieved using the `subvolume list` command.

CephMdsCpuUsageHigh

The storage metadata service (MDS) serves as file system metadata. The MDS is crucial for any file creation, rename, deletion, and update operations. By default, MDS is allocated for two or three CPUs, which does not cause issues usually when not too many metadata operations are on going. When the metadata operation load increases enough to trigger this alert, it means that the default CPU allocation is unable to cope with the load. In such a scenario, increase the CPU allocation or run multiple active MDS servers.

Impact: High

Diagnosis

1. Click Workloads > Pods.
2. Select the corresponding MDS pod and click the Metrics tab.

On the Metrics tab, you can see the allocated and used CPU. When the CPU usage exceeds 67% of the allocated CPU for 6 hours, the CephMdsCpuUsageHigh alert is triggered.

Mitigation

Do a vertical or a horizontal scaling of CPU. For more information, see the Description and Runbook section of the alert.

Run the following command to set the number of allocated CPU for MDS. For example, referring eight CPUs in the following command:

```
oc patch -n openshift-storage storagecluster ocs-storagecluster \
   --type merge \
   --patch '{"spec": {"resources": {"mds": {"limits": {"cpu": "8"},
   "requests": {"cpu": "8"}}}}}'
```

Run the following command to run multiple active MDS servers:

```
oc patch -n openshift-storage storagecluster ocs-storagecluster\
   --type merge \
   --patch '{"spec": {"managedResources": {"cephFilesystems":{"activeMetadataServers": 2}}}}'
```

Ensure that enough CPUs are allocated to run multiple active MDS servers, depending on the load.

Important: Always increase the `activeMetadataServers` by `1`. The scaling of `activeMetadataServers` works only if you have more than one PersistentVolume (PV). If only one PV is causing CPU load, increase the allocated CPU as described in the mitigation section.

CephMdsMissingReplicas

Minimum required replicas for the storage metadata service (MDS) are not available. MDS is responsible for filing metadata. Degradation of the MDS service can affect how the storage cluster works (related to the CephFS storage class) and should be fixed as soon as possible.

Impact: High

Diagnosis

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep rook-ceph-mds**
2. Set **`MYPOD`** as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
   Examine the output for a {ceph-component} that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.
pod status: NOT pending, but NOT running
Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important: If a node was assigned, check the kubelet on the node.

Mitigation

(Optional) Debugging log information
Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephMgrIsAbsent

Not having a Ceph manager runningpts the monitoring of the cluster. Persistent Volume Claim (PVC) creation and deletion requests should be resolved as soon as possible.

Impact: High

Diagnosis

Verify that the **`rook-ceph-mgr`** pod is failing, and restart if necessary. If the Ceph mgr pod restart fails, follow the general pod troubleshooting steps to resolve the issue.

1. Verify that the Ceph mgr pod is failing, using the **oc get pods | grep mgr** command.
2. Describe the Ceph mgr pod for more details, using the **oc describe pods/<pod_name>** command, where *<pod_name>* specifies the **`rook-ceph-mgr`** pod name from the previous step.
3. Analyze the errors related to resource issues.
4. Delete the pod, and wait for the pod to restart, using the **oc get pods | grep mgr** command.

Use the following steps for general pod troubleshooting:

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep rook-ceph-mgr**
2. Set **`MYPOD`** as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
   Examine the output for a {ceph-component} that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.
pod status: NOT pending, but NOT running
Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important: If a node was assigned, check the kubelet on the node.

Mitigation

(Optional) Debugging log information
Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephMgrIsMissingReplicas

To resolve this alert, you need to determine the cause of the disappearance of the Ceph manager and restart if necessary.

Impact: High

Diagnosis

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep rook-ceph-mgr**
2. Set `MYPOD` as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
   Examine the output for a {ceph-component} that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.
pod status: NOT pending, but NOT running
Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important: If a node was assigned, check the kubelet on the node.

Mitigation

(Optional) Debugging log information
Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephMonHighNumberOfLeaderChanges

In a Ceph cluster there is a redundant set of monitor pods that store critical information about the storage cluster. Monitor pods synchronize periodically to obtain information about the storage cluster. The first monitor pod to get the most updated information becomes the leader, and the other monitor pods start their synchronization process after asking the leader. A problem in network connection or another kind of problem in one or more monitor pods produces an unusual change of the leader. This situation can negatively affect the storage cluster performance.

Impact: Medium
Important: Check for any network issues. If there is a network issue, you need to escalate to the Fusion Data Foundation team before you proceed with any of the following troubleshooting steps.

Diagnosis

Use one of the following to help gather more information and diagnose the issue:

- Print the logs of the affected monitor pod to gather more information about the issue, using the **oc logs <rook-ceph-mon-X-yyyy> -n openshift-storage** command, where *<rook-ceph-mon-X-yyyy>* specifies the name of the affected monitor pod.
- Use the Openshift Web console to open the logs of the affected monitor pod. More information about possible causes is reflected in the log.

Use the following steps for general pod troubleshooting:

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep {ceph-component}**
2. Set `MYPOD` as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
   Examine the output for a {ceph-component} that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.
pod status: NOT pending, but NOT running
Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important: If a node was assigned, check the kubelet on the node.

Mitigation

(Optional) Debugging log information
Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephMonQuorumAtRisk

Multiple MONs work together to provide redundancy. Each of the MONs keeps a copy of the metadata. The cluster is deployed with 3 MONs, and requires 2 or more MONs to be up and running for quorum and for the storage operations to run. If quorum is lost, access to data is at risk.

Impact: High

Diagnosis

Restore the Ceph MON Quorum. For more information, see *Restoring ceph-monitor quorum in Fusion Data Foundation* in the Troubleshooting guide.

If the restoration of the Ceph MON Quorum fails, follow the general pod troubleshooting to resolve the issue.

Use the following steps for general pod troubleshooting:

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep rook-ceph-mon**
2. Set `MYPOD` as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
   Examine the output for a {ceph-component} that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.
pod status: NOT pending, but NOT running
Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important: If a node was assigned, check the kubelet on the node.

Mitigation

(Optional) Debugging log information
Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephMonQuorumLost

In a Ceph cluster there is a redundant set of monitor pods that store critical information about the storage cluster. Monitor pods synchronize periodically to obtain information about the storage cluster. The first monitor pod to get the most updated information becomes the leader, and the other monitor pods start their synchronization process after asking the leader. A problem in network connection or another kind of problem in one or more monitor pods produces an unusual change of the leader. This situation can negatively affect the storage cluster performance.

Impact: High
Note: Check for any network issues. If there is a network issue, you need to escalate to the Fusion Data Foundation team before you proceed with any of the following troubleshooting steps.

Diagnosis

Restore the Ceph MON Quorum. For more information, see Restoring ceph-monitor quorum in Fusion Data Foundation.

If the restoration of the Ceph MON Quorum fails, follow the general pod troubleshooting to resolve the issue.

Use the following steps for general pod troubleshooting:

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep {ceph-component}**
2. Set `MYPOD` as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
   Examine the output for a {ceph-component} that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.
pod status: NOT pending, but NOT running
Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important: If a node was assigned, check the kubelet on the node.

Mitigation

(Optional) Debugging log information

Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephMonVersionMismatch

Typically this alert triggers during an upgrade that is taking a long time.

Impact: Medium

Diagnosis

Check the `ocs-operator` subscription status and the operator pod health to check if an operator upgrade is in progress.

1. Check the `ocs-operator` subscription health.

   ```
   oc get sub $(oc get pods -n openshift-storage | grep -v ocs-operator) -n openshift-storage -o json | jq .status.conditions
   ```

   The status condition types are `CatalogSourcesUnhealthy`, `InstallPlanMissing`, `InstallPlanPending`, and `InstallPlanFailed`. The status for each type should be `False`.

   Example output:

   ```
   [
     {
       "lastTransitionTime": "2021-01-26T19:21:37Z",
       "message": "all available catalogsources are healthy",
       "reason": "AllCatalogSourcesHealthy",
       "status": "False",
       "type": "CatalogSourcesUnhealthy"
     }
   ]
   ```

   The example output shows a `False` status for type `CatalogSourcesUnHealthly`, which means that the catalog sources are healthy.
2. Check the OCS operator pod status to see if there is an OCS operator upgrading in progress.

   ```
   oc get pod -n openshift-storage | grep ocs-operator OCSOP=$(oc get pod -n openshift-storage -o custom-
   columns=POD:.metadata.name --no-headers | grep ocs-operator) echo $OCSOP oc get pod/${OCSOP} -n openshift-storage oc
   describe pod/${OCSOP} -n openshift-storage
   ```

   If you determine that the `ocs-operator` is in progress, wait for 5 minutes and this alert should resolve itself. If you have waited or see a different error status condition, continue troubleshooting.

Mitigation

(Optional) Debugging log information

Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephNodeDown

A node running Ceph pods is down. While storage operations continue to function as Ceph is designed to deal with a node failure, it is recommended to resolve the issue to minimize the risk of another node going down and affecting storage functions.

Impact: Medium

Diagnosis

1. List all the pods that are running and failing:

   ```
   oc -n openshift-storage get pods
   ```

   Important: Ensure that you meet the IBM Fusion Data Foundation resource requirements so that the Object Storage Device (OSD) pods are scheduled on the new node. This may take a few minutes as the Ceph cluster recovers data for the failing but now recovering OSD. To watch this recovery in action, ensure that the OSD pods are correctly placed on the new worker node.
2. Check if the OSD pods that were previously failing are now running:

   ```
   oc -n openshift-storage get pods
   ```

   If the previously failing OSD pods have not been scheduled, use the **describe** command and check the events for reasons the pods were not rescheduled.
3. Describe the events for the failing OSD pod:

   ```
   oc -n openshift-storage get pods | grep osd
   ```

4. Find the one or more failing OSD pods:

`oc -n openshift-storage describe pods/<osd_podname_ from_the_ previous step>`

In the events section look for the failure reasons, such as the resources are not being met.

- (Optional) Use the `rook-ceph-toolbox` to watch the recovery. This step is helpful for large Ceph clusters. To access the toolbox, run the following command:

```
TOOLS_POD=$(oc get pods -n openshift-storage -l app=rook-ceph-tools -o name)
oc rsh -n openshift-storage $TOOLS_POD
```

- From the rsh command prompt, run the **ceph status** command and watch for *recovery* under the I/O section.

5. Determine if there are failed nodes.
   a. Get the list of worker nodes, and check for the node status:

```
oc get nodes --selector='node-role.kubernetes.io/worker','!node-role.kubernetes.io/infra'
```

   b. Describe the node which is of the *NotReady* status to get more information about the failure:

```
oc describe node <node_name>
```

Mitigation

(Optional) Debugging log information

Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephOSDCriticallyFull

One of the Object Storage Devices (OSDs) is critically full. Expand the cluster immediately.

Impact: High

Diagnosis

Deleting data to free up storage space

You can delete data, and the cluster resolves the alert through self healing processes.

Important: This is only applicable to IBM Fusion Data Foundation clusters that are near or full but not in read-only mode. Read-only mode prevents any changes that include deleting data, that is, deletion of Persistent Volume Claim (PVC), Persistent Volume (PV) or both.

Expanding the storage capacity (current storage size is less than 1 TB)

You must first assess the ability to expand. For every 1 TB of storage added, the cluster needs to have 3 nodes each with a minimum available 2 vCPUs and 8 GiB memory.

You can increase the storage capacity to 4 TB through the add-on, and the cluster resolves the alert through self healing processes. If the minimum vCPU and memory resource requirements are not met, you need to add 3 additional worker nodes to the cluster.

Mitigation

If your current storage size is equal to 4 TB, contact IBM Support.

(Optional) Debugging log information

Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephOSDDiskNotResponding

A disk device is not responding. Check whether all the Object Storage Devices (OSDs) are up and running.

Impact: Medium

Diagnosis

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep rook-ceph**
2. Set `MYPOD` as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

```
 Examine the output for a rook-ceph that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready

Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.

pod status: NOT pending, but NOT running

Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important:

- If a node was assigned, check the kubelet on the node.
- If the basic health of the running pods, node affinity and resource availability on the nodes are verified, run the Ceph tools to get the status of the storage components.

Mitigation

(Optional) Debugging log information
Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephOSDDiskUnavailable

A disk device is not accessible on one of the hosts and its corresponding Object Storage Device (OSD) is marked out by the Ceph cluster. This alert is raised when a Ceph node fails to recover within 10 minutes.

Impact: High

Diagnosis

Determine the failed node

1. Get the list of worker nodes, and check for the node status:

   ```
   oc get nodes --selector='node-role.kubernetes.io/worker','!node-role.kubernetes.io/infra'
   ```

2. Describe the node which is of *NotReady* status to get more information on the failure, using the following command:

   ```
   oc describe node <node_name>
   ```

CephOSDFlapping

A storage daemon has restarted 5 times in the last 5 minutes. Check the pod events or Ceph status to find out the cause.

Impact: High

Diagnosis

Follow the steps, as detailed in Flapping OSDs within the IBM Storage Ceph documentation.
Use the following steps for general pod troubleshooting:

pod status: pending

1. Check for resource issues, pending Persistent Volume Claims (PVCs), node assignment, and kubelet problems, using the following commands:
   - **oc project openshift-storage**
   - **oc get pod | grep rook-ceph**
2. Set **MYPOD** as the variable for the pod that is identified as the problem pod, specifying the name of the pod that is identified as the problem pod for *<pod_name>*:

   ```
    Examine the output for a rook-ceph that is in the pending state, not running or not ready
   MYPOD=<pod_name>
   ```

3. Look for the resource limitations or pending PVCs. Otherwise, check for the node assignment, using the **oc get pod/${MYPOD} -o wide** command.

pod status: NOT pending, running, but NOT ready
Check the readiness of the probe, using the **oc describe pod/${MYPOD}** command.
pod status: NOT pending, but NOT running
Check for application or image issues, using the **oc logs pod/${MYPOD}** command.
Important:

- If a node was assigned, check the kubelet on the node.
- If the basic health of the running pods, node affinity and resource availability on the nodes are verified, run the Ceph tools to get the status of the storage components.

Mitigation

(Optional) Debugging log information
Run the following command to gather the debugging information for the Ceph cluster:

```
oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6
```

CephOSDNearFull

Utilization of back-end storage device Object Storage Device (OSD) has crossed 75% on a host.

Impact: High

Mitigation

Free up some space in the cluster, expand the storage cluster, or contact IBM Support.
For more information on scaling storage, see Scaling out storage capacity.

CephOSDSlowOps

An Object Storage Device (OSD) with slow requests is every OSD that is not able to service the I/O operations per second (IOPS) in the queue within the time defined by the osd_op_complaint_time parameter. By default, this parameter is set to 30 seconds.

Impact: Medium

Diagnosis

More information about the slow requests can be obtained using the Openshift console.

Access the OSD pod terminal, and run the following commands:

1. `$ ceph daemon osd.<id> ops`
2. `$ ceph daemon osd.<id> dump_historic_ops`

Note: The number of the OSD is seen in the pod name. For example, in `rook-ceph-osd-0-5d86d4d8d4-zlqkx`, `<0>` is the OSD.

Mitigation

The main causes of the OSDs having slow requests are:

- Problems with the underlying hardware or infrastructure, such as, disk drives, hosts, racks, or network switches. Use the Openshift monitoring console to find the alerts or errors about cluster resources. This can give you an idea about the root cause of the slow operations in the OSD.
- Problems with the network. These problems are usually connected with flapping OSDs. See Flapping OSDs within the IBM Storage Ceph documentation..
- If it is a network issue, escalate to IBM Support.
- System load. Use the Openshift console to review the metrics of the OSD pod and the node which is running the OSD. Adding or assigning more resources can be a possible solution.

CephOSDVersionMismatch

Typically this alert triggers during an upgrade that is taking a long time.

Impact: Medium

Diagnosis

Check the `ocs-operator` subscription status and the operator pod health to check if an operator upgrade is in progress.

1. Check the `ocs-operator` subscription health.

   ```
   oc get sub $(oc get pods -n openshift-storage | grep -v ocs-operator) -n openshift-storage -o json | jq .status.conditions
   ```

   The status condition types are `CatalogSourcesUnhealthy`, `InstallPlanMissing`, `InstallPlanPending`, and `InstallPlanFailed`. The status for each type should be `False`.

   Example output:

   ```
   [
     {
       "lastTransitionTime": "2021-01-26T19:21:37Z",
       "message": "all available catalogsources are healthy",
       "reason": "AllCatalogSourcesHealthy",
       "status": "False",
       "type": "CatalogSourcesUnhealthy"
     }
   ]
   ```

   The example output shows a `False` status for type `CatalogSourcesUnHealthly`, which means that the catalog sources are healthy.
2. Check the OCS operator pod status to see if there is an OCS operator upgrading in progress.

   ```
   oc get pod -n openshift-storage | grep ocs-operator OCSOP=$(oc get pod -n openshift-storage -o custom-columns=POD:.metadata.name --no-headers | grep ocs-operator) echo $OCSOP oc get pod/${OCSOP} -n openshift-storage oc describe pod/${OCSOP} -n openshift-storage
   ```

If you determine that the `ocs-operator` is in progress, wait for 5 minutes and this alert should resolve itself. If you have waited or see a different error status condition, continue troubleshooting.

CephPGRepairTakingTooLong

Self-healing operations are taking too long.

Impact: High

Diagnosis

Check for inconsistent Placement Groups (PGs), and repair them. For more information, see the Red Hat Knowledgebase solution Handle Inconsistent Placement Groups in Ceph.

CephPoolFull

One or more storage pools have reached 85% capacity on storage cluster. Write operations to affected pools are blocked.

Impact: Critical

Diagnosis

Scaling storage
: You need to immediately add storage capacity. For more information, see Scaling out storage capacity.

Mitigation

Deleting information
: If it is not possible to scale out the cluster, you need to delete information to free up some space.

CephPoolNearFull

One or more storage pools are approaching capacity limits of 75% on storage cluster. Consider adding more storage capacity.

Impact: Warning

Diagnosis

Scaling storage
: You need to add storage capacity. For more information, see Scaling out storage capacity.

Mitigation

Deleting information
: If it is not possible to scale out the cluster, you need to delete information to free up some space.

CephPoolQuotaBytesCriticallyExhausted

One or more pools has reached, or is very close to reaching, its quota. The threshold to trigger this error condition is controlled by the `mon_pool_quota_crit_threshold` configuration option.

Impact: High

Mitigation

Adjust the pool quotas.
Run the following commands to fully remove or adjust the pool quotas up or down:

- `ceph osd pool set-quota <pool> max_bytes <bytes>`
- `ceph osd pool set-quota <pool> max_objects <objects>`

Setting the quota value to `0` disables the quota.

CephPoolQuotaBytesNearExhaustion

One or more pools is approaching a configured fullness threshold. One threshold that can trigger this warning condition is the `mon_pool_quota_warn_threshold` configuration option.

Impact: High

Mitigation

Adjust the pool quotas.
Run the following commands to fully remove or adjust the pool quotas up or down:

- `ceph osd pool set-quota <pool> max_bytes <bytes>`
- `ceph osd pool set-quota <pool> max_objects <objects>`

Setting the quota value to `0` disables the quota.

CephXattrSetLatency

Ceph Metadata Server (MDS) is experiencing high latency when setting extended attributes (xattr) on files. The alert fires when the average setxattr operation latency exceeds 30 milliseconds over a 5-minute period.

Impact: Medium

Diagnosis

1. Check MDS status and health.
2. Go to the OpenShift console.
3. Access the Workloads section and select the MDS pod.
4. Click the Metrics tab to review CPU and memory usage.
5. Check the network issues as network latency between MDS and OSDs can cause slow metadata operations.
6. Check OSD performance, as slow OSD operations can cascade into MDS latency issues.

Mitigation

You need to increase MDS CPU resources, MDS cache memory, scale out with multiple active MDS, use faster metadata storage, address network issues, address underlying OSD issue, or restart MDS. For more information, see the Description and Runbook section of the alert.

OSDCPULoadHigh

OSD is a critical component in Ceph storage, responsible for managing data placement and recovery. High CPU usage in the OSD container suggests increased processing demands, potentially leading to degraded storage performance.

Impact: High

Diagnosis

1. Go to the Kubernetes dashboard or equivalent.
2. Access the Workloads section and select the relevant pod associated with the OSD alert.
3. Click the Metrics tab to view CPU metrics for the OSD container.
4. Verify that the CPU usage exceeds 80% over a significant period (as specified in the alert configuration).

Mitigation

If the OSD CPU usage is consistently high, consider taking the following steps:

1. Evaluate the overall storage cluster performance and identify the OSDs contributing to high CPU usage.
2. Increase the number of OSDs in the cluster by adding more new storage devices in the existing nodes or adding new nodes with new storage devices. Review the Scaling out storage capacity for instructions to help distribute the load and improve overall system performance.

PersistentVolumeUsageCritical

A Persistent Volume Claim (PVC) is nearing its full capacity and may lead to data loss if not attended to in a timely manner.

Impact: High

Mitigation

- Expand the PVC size to increase the capacity.
    1. Log in to the OpenShift Web Console.
    2. Go to Storage > PersistentvolumeClaim.
    3. Select openshift-storage from the Project drop-down list.
    4. On the PVC you want to expand, go to Action menu (⋮) > Expand PVC.
    5. Update the Total size to the desired size.
    6. Click Expand.
- Alternatively, you can delete unnecessary data that may be taking up space.

- `ceph osd pool set-quota <pool> max_bytes <bytes>`
- `ceph osd pool set-quota <pool> max_objects <objects>`

Setting the quota value to `0` disables the quota.

PersistentVolumeUsageNearFull

A Persistent Volume Claim (PVC) is nearing its full capacity and may lead to data loss if not attended to in a timely manner.

Impact: High

Mitigation

- Expand the PVC size to increase the capacity.
    1. Log in to the OpenShift Web Console.
    2. Go to Storage > PersistentvolumeClaim.
    3. Select openshift-storage from the Project drop-down list.
    4. On the PVC you want to expand, go to Action menu (⋮) > Expand PVC.
    5. Update the Total size to the desired size.
    6. Click Expand.
- Alternatively, you can delete unnecessary data that may be taking up space.

- `ceph osd pool set-quota <pool> max_bytes <bytes>`
- `ceph osd pool set-quota <pool> max_objects <objects>`

Setting the quota value to `0` disables the quota.

NooBaaSystemCapacityWarning85

The NooBaa system capacity is calculated by aggregating the total storage capacity of all backingstore resources. Namespacestore resources are not included in this calculation. This alert is triggered when the total used space across the backingstore resources exceeds 85% of the total available capacity.

Impact: Warning

Diagnosis

Depending on the type of backingstore resource, the diagnosis may vary. For backingstores based on a public cloud provider, the free space is always assumed to be 1 Petabyte, and in such cases the alert is not relevant. For backingstores that use a compatible S3 service, check the usage and available space of the target bucket used by the backingstore. For PV-Pool backingstores, review the usage and available space of the PersistentVolumes (PVs) used by the pv-pool pods.

Mitigation

For public cloud providers, no action is required. For compatible S3 services, check the usage and available space of the target bucket used by the backingstore. If the usage is approaching the bucket's capacity, consider increasing its allocated capacity. For PV-Pool backingstores, review the usage and available space of the PVs used by the pv-pool pods. If the usage is nearing capacity, consider increasing the number of volumes in the backingstore specification to add new PVs to the pool. The number of volumes cannot be reduced later.

NooBaaSystemCapacityWarning95

The NooBaa system capacity is calculated by aggregating the total storage capacity of all backingstore resources. Namespacestore resources are not included in this calculation. This alert is triggered when the total used space across the backingstore resources exceeds 95% of the total available capacity.

Impact: Warning

Diagnosis

The diagnosis depends on the type of backingstore resource in use. For backingstores based on a public cloud provider, the free space is always assumed to be 1 Petabyte, making this alert not applicable. For backingstores configured with a compatible S3 service, check the usage and available space of the target bucket used by the backingstore. For PV-Pool backingstores, review the usage and available capacity of the PersistentVolumes (PVs) used by the pv-pool pods.

Mitigation

For public cloud providers, no action is required. For compatible S3 services, check the usage and available space of the target bucket used by the backingstore. If usage is nearing the bucket's capacity, consider increasing the bucket's allocated capacity. For PV-Pool backingstores, review the usage and available space of the PVs used by the pv-pool pods. If the usage is approaching capacity, increase the number of volumes in the backingstore specification to add new PVs to the pool.

Note: The number of volumes cannot be reduced later.

NooBaaSystemCapacityWarning100

The NooBaa system capacity is calculated by aggregating the total storage capacity of all backingstore resources. Namespacestore resources are not included in this calculation. This alert is triggered when the total used space across the backingstore resources reaches 100% of the available capacity.

Impact: High

Diagnosis

Depending on the type of backingstore resource, the diagnosis may differ. For backingstores based on a public cloud provider, the free space is always assumed to be 1 Petabyte, making the alert not applicable. For backingstores that use a compatible S3 service, check the usage and available space of the target bucket associated with the backingstore. For PV-Pool backingstores, review the usage and available space of the PersistentVolumes (PVs) used by the pv-pool pods.

Mitigation

For public cloud providers, no action is required. For compatible S3 services, check the usage and available space of the target bucket used by the backingstore. If the usage is approaching its capacity limit, consider increasing the bucket's capacity. For PV-Pool backingstores, review the usage and available space of the PVs used by the pv-pool pods. If additional capacity is needed, increase the number of volumes in the backingstore specification to add new PVs to the pool. The number of volumes cannot be reduced once configured.

Resolving NooBaa Bucket Error State

Use this information to resolve a NooBaa Bucket Error State.

Procedure

1. From the OpenShift Web Console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. Click the Object tab.
4. In the Details card, click the link under System Name field.
5. In the left pane, click Buckets option and search for the bucket in error state. If the bucket in error state is a namespace bucket, be sure to click the Namespace Buckets pane.
6. Click it’s Bucket Name. Error encountered in bucket is displayed.
7. Depending on the specific error of the bucket, perform one or both of the following:

   For space related errors
   a. In the left pane, click Resources option.
   b. Click the resource in error state.
   c. Scale the resource by adding more agents.

   For resource health errors
   a. In the left pane, click Resources option.
   b. Click the resource in error state.
   c. Connectivity error means the backing service is not available and needs to be restored.
   d. For access/permissions errors, update the connection’s Access Key and Secret Key.

Resolving NooBaa Bucket Exceeding Quota State

Use this information to resolve a `A NooBaa Bucket Is In Exceeding Quota State` error.

About this task

To resolve **A NooBaa Bucket Is In Exceeding Quota State** error perform one of the following:

- Cleanup some of the data on the bucket.
- Increase the bucket quota by performing the following steps.

Procedure

1. From the OpenShift Web Console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. Click the Object tab.
4. In the Details card, click the link under System Name field.
5. In the left pane, click Buckets option and search for the bucket in error state.
6. Click its Bucket Name. Error encountered in bucket is displayed.
7. Click Bucket Policies > Edit Quota and increase the quota.

Resolving NooBaa Bucket Capacity or Quota State

Use this information to resolve a NooBaa Bucket Capacity or Quota State.

Procedure

1. From the OpenShift Web Console, go to Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the pop up that appears.
3. Click the Object tab.
4. In the Details card, click the link under System Name field.
5. In the left pane, click the Resources option and search for the PV pool resource.
6. For the PV pool resource with low capacity status, click on its Resource Name.
7. Edit the pool configuration and increase the number of agents.

Recovering pods

Use this information to recover a pod.

About this task

When a first node (for example **`NODE1`**) goes to *NotReady* state because of some issue, the hosted pods that are using PVC with ReadWriteAfter (RWO) access mode try to move to the second node (for example **`NODE2`**) but get stuck due to multi-attach error. In such a case, you can recover MON, OSD, and application pods by using the following steps.

Procedure

1. Power off **`NODE1`** (from AWS or vSphere side) and ensure that **`NODE1`** is completely down.
2. Force delete the pods on **`NODE1`** by using the following command:

```
oc delete pod <pod-name> --grace-period=0 --force
```

Recovering from EBS volume detach

When an OSD or MON elastic block storage (EBS) volume where the OSD disk resides is detached from the worker Amazon EC2 instance, the volume gets reattached automatically within one or two minutes. However, the OSD pod gets into a **`CrashLoopBackOff`** state.

Recover and bring back the pod to *Running* state, by restarting the EC2 instance.

Enabling debug logs for rook-ceph-operator

Enable the debug logs for the rook-ceph-operator to obtain information about failures that help in troubleshooting issues.

Procedure

1. Edit the configmap of the rook-ceph-operator.

```
oc edit configmap rook-ceph-operator-config
```

2. Add the ROOK_LOG_LEVEL: DEBUG parameter in the rook-ceph-operator-config yaml file to enable the debug logs for rook-ceph-operator.

```
…
data:
  # The logging level for the operator: INFO | DEBUG
  ROOK_LOG_LEVEL: DEBUG
```

The rook-ceph-operator logs now consist of the debug information.

Disabling debug logs for rook-ceph-operator

Disable the debug logs for the rook-ceph-operator.

Procedure

1. Edit the configmap of the rook-ceph-operator.

```
oc edit configmap rook-ceph-operator-config
```

2. Add the ROOK_LOG_LEVEL: INFO parameter in the rook-ceph-operator-config yaml file to disable the debug logs for rook-ceph-operator.

```
…
data:
  # The logging level for the operator: INFO | DEBUG
  ROOK_LOG_LEVEL: INFO
```

Resolving low Ceph monitor count alert

The **CephMonLowNumber** alert is displayed in the notification panel or Alert Center of the OpenShift Web Console to indicate the low number of Ceph monitor count when your internal mode deployment has five or more nodes, racks, or rooms, and when there are five or more number of failure domains in the deployment. You can increase the Ceph monitor count to improve the availability of cluster.

Procedure

1. In the **CephMonLowNumber** alert of the notification panel or Alert Center of Red Hat® OpenShift® Container Platform web console, click Configure.
2. In the Configure Ceph Monitor pop up, click Update count.
   In the pop up, the recommended monitor count depending on the number of failure zones is shown.
3. In the Configure CephMon pop up, update the monitor count value based on the recommended value and click Save changes.

Troubleshooting unhealthy blocklisted nodes

If an RBD client on a Kubernetes node is blocklisted by Ceph, causing `CreateContainerError` pod states and potentially read-only PVCs, taint the node to evict pods or reboot the node to clear the blocklist.

ODFRBDClientBlocked

Meaning	This alert indicates that an RBD client might be blocked by Ceph on a specific node within your Kubernetes cluster. The blocklisting occurs when the `ocs_rbd_client_blocklisted metric` reports a value of 1 for the node. Additionally, there are pods in a `CreateContainerError` state on the same node. The blocklisting can potentially result in the filesystem for the Persistent Volume Claims (PVCs) using RBD becoming read-only. It is crucial to investigate this alert to prevent any disruption to your storage cluster.
Impact	High

Diagnosis

The blocklisting of an RBD client can occur due to several factors, such as network or cluster slowness. In certain cases, the exclusive lock contention among three contending clients (workload, mirror daemon, and manager/scheduler) can lead to the blocklist.

Mitigation

1. Taint the blocklisted node: In Kubernetes, consider tainting the node that is blocklisted to trigger the eviction of pods to another node. This approach relies on the assumption that the unmounting/unmapping process progresses gracefully. After the pods have been successfully evicted, the blocklisted node can be untainted, allowing the blocklist to be cleared. The pods can then be moved back to the untainted node.
2. Reboot the blocklisted node: If tainting the node and evicting the pods do not resolve the blocklisting issue, a reboot of the blocklisted node can be attempted. This step may help alleviate any underlying issues causing the blocklist and restore normal functionality.

Important: Investigating and resolving the blocklist issue promptly is essential to avoid any further impact on the storage cluster.

Checking for Local Storage Operator deployments

IBM Fusion Data Foundation clusters with Local Storage Operator are deployed using local storage devices. To find out if your existing cluster with Fusion Data Foundation was deployed using local storage devices, use the following procedure:

Before you begin

- Fusion Data Foundation is installed and running in the `openshift-storage` namespace.

About this task

By checking the storage class associated with your Fusion Data Foundation cluster's persistent volume claims (PVCs), you can tell if your cluster was deployed using local storage devices.

Procedure

1. Check the storage class associated with Fusion Data Foundation cluster's PVCs with the following command:

   ```
   oc get pvc -n openshift-storage
   ```

2. Check the output.
   For clusters with Local Storage Operator, the PVCs associated with `ocs-deviceset` use the storage class `localblock`. The output looks similar to the following:

   ```
   NAME                      STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
   AGE
   db-noobaa-db-0            Bound    pvc-d96c747b-2ab5-47e2-b07e-1079623748d8   50Gi       RWO            ocs-
   storagecluster-ceph-rbd   114s
   ocs-deviceset-0-0-lzfrd   Bound    local-pv-7e70c77c                          1769Gi     RWO            localblock
   2m10s
   ocs-deviceset-1-0-7rggl   Bound    local-pv-b19b3d48                          1769Gi     RWO            localblock
   2m10s
   ocs-deviceset-2-0-znhk8   Bound    local-pv-e9f22cdc                          1769Gi     RWO            localblock
   2m10s
   ```

Removing failed or unwanted Ceph Object Storage devices

Failed or unwanted Ceph Object Storage Devices (OSDs) reduce storage cluster reliability and must be removed after verifying Ceph health, using the procedure appropriate for dynamically provisioned or local storage deployments.

The failed or unwanted Ceph OSDs (Object Storage Devices) affects the performance of the storage infrastructure. Hence, to improve the reliability and resilience of the storage cluster, you must remove the failed or unwanted Ceph OSDs.

If you have any failed or unwanted Ceph OSDs to remove:

1. Verify the Ceph health status.

   For more information see: Verifying Ceph cluster is healthy.

2. Based on the provisioning of the OSDs, remove failed or unwanted Ceph OSDs.

   See:

   - Removing failed or unwanted Ceph OSDs in dynamically provisioned Red Hat OpenShift Data Foundation.
   - Removing failed or unwanted Ceph OSDs provisioned using local storage devices.

- **Verifying Ceph cluster is healthy**
  Verify Ceph cluster health from the OpenShift Web Console by checking the Storage Cluster status card and confirming the cluster information is displayed in the Block and File dashboard.
- **Removing failed or unwanted Ceph OSDs in dynamically provisioned Red Hat OpenShift Data Foundation**
  Remove failed or unwanted Ceph OSDs in a dynamically provisioned deployment by scaling down the OSD, deleting its prepare pod, running the OSD removal job, and deleting the OSD deployment.
- **Removing failed or unwanted Ceph OSDs provisioned using local storage devices**
  Remove failed or unwanted Ceph OSDs provisioned with local storage by scaling down the OSD, running the removal job, deleting PVC and PV resources, removing the device symlink, and verifying the removal.
- **Troubleshooting the error cephosd:osd.0 is NOT ok to destroy while removing failed or unwanted Ceph OSDs**
  If the `ocs-osd-removal-job` reports `cephosd:osd.0 is NOT ok to destroy`, re-run the OSD removal job with the `FORCE_OSD_REMOVAL=true` option, provided all placement groups are in an active state.

Verifying Ceph cluster is healthy

Verify Ceph cluster health from the OpenShift Web Console by checking the Storage Cluster status card and confirming the cluster information is displayed in the Block and File dashboard.

About this task

Storage health is visible on the **Block** and **File** and **Object** dashboards.

Procedure

1. In the OpenShift Web Console, click Storage > Data Foundation.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
3. In the **Status** card of the **Block and File** tab, verify that *Storage Cluster* has a green tick.
4. In the **Details** card, verify that the cluster information is displayed.

Removing failed or unwanted Ceph OSDs in dynamically provisioned Red Hat OpenShift Data Foundation

Remove failed or unwanted Ceph OSDs in a dynamically provisioned deployment by scaling down the OSD, deleting its prepare pod, running the OSD removal job, and deleting the OSD deployment.

Before you begin

- Check if Ceph is healthy. For more information, see Verifying Ceph cluster is healthy.
- Ensure that no alerts are fired or any rebuilding process is in progress.

About this task

Follow the steps in the procedure to remove the failed or unwanted Ceph OSDs in dynamically provisioned IBM Fusion Data Foundation.
Important: Scaling down of cluster is supported only with the help of the IBM Support team.
Warning:

- Removing an OSD when the Ceph component is not in a healthy state can result in data loss.
- Removing two or more OSDs at the same time results in data loss.

Procedure

1. Scale down the OSD deployment.

   ```
   # oc scale deployment rook-ceph-osd-<osd-id> --replicas=0
   ```

2. Get the `osd-prepare` pod for the Ceph OSD to be removed.

   ```
   # oc get deployment rook-ceph-osd-<osd-id> -oyaml | grep ceph.rook.io/pvc
   ```

3. Delete the `osd-prepare` pod.

   ```
   # oc delete -n openshift-storage pod rook-ceph-osd-prepare-<pvc-from-above-command>-<pod-suffix>
   ```

4. Remove the failed OSD from the cluster.

   ```
   # failed_osd_id=<osd-id>
   ```

   ```
   # oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=$<failed_osd_id> | oc create -f -
   ```

5. Verify that the OSD is removed successfully by checking the logs.

   ```
   # oc logs -n openshift-storage ocs-osd-removal-$<failed_osd_id>-<pod-suffix>
   ```

6. Optional: If you get an error as `FAILED_OSD_ID` from the `ocs-osd-removal-job` pod in OpenShift Container Platform, see Troubleshooting the error cephosd:osd.0 is NOT ok to destroy while removing failed or unwanted Ceph OSDs.
7. Delete the OSD deployment.

   ```
   oc delete deployment rook-ceph-osd-<osd-id>
   ```

What to do next

Verification step

- To check if the OSD is deleted successfully, run:

  ```
  oc get pod -n openshift-storage ocs-osd-removal-$<failed_osd_id>-<pod-suffix>
  ```

  This command must return the status as `Completed`.

Removing failed or unwanted Ceph OSDs provisioned using local storage devices

Remove failed or unwanted Ceph OSDs provisioned with local storage by scaling down the OSD, running the removal job, deleting PVC and PV resources, removing the device symlink, and verifying the removal.

About this task

You can remove failed or unwanted Ceph provisioned using local storage devices by following the steps in the procedure.
Important: Scaling down of cluster is supported only with the help of the Red Hat support team.
Warning:

- Removing an OSD when the Ceph component is not in a healthy state can result in data loss.
- Removing two or more OSDs at the same time results in data loss.

Before you begin

- Check if Ceph is healthy. For more information see Verifying Ceph cluster is healthy.
- Ensure no alerts are firing or any rebuilding process is in progress.

Procedure

1. Forcibly, mark the OSD down by scaling the replicas on the OSD deployment to 0. You can skip this step if the OSD is already down due to failure.

   ```
   oc scale deployment rook-ceph-osd-<osd-id> --replicas=0
   ```

2. Remove the failed OSD from the cluster.

   ```
   # failed_osd_id=<osd_id>
   ```

   ```
   # oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=$<failed_osd_id> | oc create -f -
   ```

3. Verify that the OSD is removed successfully by checking the logs.

   ```
   # oc logs -n openshift-storage ocs-osd-removal-$<failed_osd_id>-<pod-suffix>
   ```

4. Optional: If you get an error as `cephosd:osd.0 is NOT ok to destroy` from the **ocs-osd-removal-job** pod in OpenShift Container Platform, see Troubleshooting the error `cephosd:osd.0 is NOT ok to destroy` while removing failed or unwanted Ceph OSDs.
5. Delete persistent volume claim (PVC) resources associated with the failed OSD.
   a. Get the PVC associated with the failed OSD.

      ```
      oc get -n openshift-storage -o yaml deployment rook-ceph-osd-<osd-id> | grep ceph.rook.io/pvc
      ```

   b. Get the persistent volume (PV) associated with the PVC.

      ```
      oc get -n openshift-storage pvc <pvc-name>
      ```

   c. Get the failed device name.

      ```
      oc get pv <pv-name-from-above-command> -oyaml | grep path
      ```

   d. Get the prepare-pod associated with the failed OSD.

      ```
      oc describe -n openshift-storage pvc ocs-deviceset-0-0-nvs68 | grep Mounted
      ```

   e. Delete the osd-prepare pod before removing the associated PVC.

      ```
      oc delete -n openshift-storage pod <osd-prepare-pod-from-above-command>
      ```

   f. Delete the PVC associated with the failed OSD.

      ```
      oc delete -n openshift-storage pvc <pvc-name-from-step-a>
      ```

6. Remove failed device entry from the `LocalVolume custom resource`(CR).
   a. Log in to node with the failed device.

      ```
      oc debug node/<node_with_failed_osd>
      ```

   b. Record the /dev/disk/by-id/<id> for the failed device name.

      ```
      ls -alh /mnt/local-storage/localblock/
      ```

7. Optional: In case, Local Storage Operator is used for provisioning OSD, login to the machine with {osd-id} and remove the device symlink.

   ```
   oc debug node/<node_with_failed_osd>
   ```

   a. Get the OSD symlink for the failed device name.

      ```
      ls -alh /mnt/local-storage/localblock
      ```

   b. Remove the symlink

      ```
      rm /mnt/local-storage/localblock/<failed-device-name>
      ```

8. Delete the PV associated to the OSD.

   ```
   # oc delete pv <pv-name>
   ```

What to do next

- To check if the OSD is deleted successfully, run:

  ```
  #oc get pod -n openshift-storage ocs-osd-removal-$<failed_osd_id>-<pod-suffix>
  ```

  This command must return the status as **Completed**.

Troubleshooting the error cephosd:osd.0 is NOT ok to destroy while removing failed or unwanted Ceph OSDs

If the `ocs-osd-removal-job` reports `cephosd:osd.0 is NOT ok to destroy`, re-run the OSD removal job with the `FORCE_OSD_REMOVAL=true` option, provided all placement groups are in an active state.

If you get an error as `cephosd:osd.0 is NOT ok to destroy` from the **ocs-osd-removal-job** pod in OpenShift Container Platform, run the OSD removal job with **FORCE_OSD_REMOVAL** option to move the OSD to a destroyed state.

```
oc process -n openshift-storage ocs-osd-removal -p FORCE_OSD_REMOVAL=true -p FAILED_OSD_IDS=$<failed_osd_id> | oc create -f -
```

Note: You must use the **FORCE_OSD_REMOVAL** option only if all the PGs are in active state. If not, PGs must either complete the back filling or further investigated to ensure they are active.

Troubleshooting and deleting remaining resources during Uninstall

Occasionally some of the custom resources managed by an operator may remain in *Terminating* status waiting on the finalizer to complete, although you have performed all the required cleanup tasks. In such an event you need to force the removal of such resources. If you do not do so, the resources remain in the *Terminating* state even after you have performed all the uninstall steps.

Procedure

1. Check if the openshift-storage namespace is stuck in *Terminating* state upon deletion.

   ```
   oc get project -n <namespace>
   ```

   Output:

   ```
   NAME DISPLAY NAME STATUS
   openshift-storage Terminating
   ```

2. Check for the `NamespaceFinalizersRemaining` and `NamespaceContentRemaining` messages in the `STATUS` section of the command output and perform the next step for each of the listed resources.

   ```
   oc get project openshift-storage -o yaml
   ```

   Example output :

   ```
   status:
     conditions:
     - lastTransitionTime: "2020-07-26T12:32:56Z"
       message: All resources successfully discovered
       reason: ResourcesDiscovered
       status: "False"
       type: NamespaceDeletionDiscoveryFailure
     - lastTransitionTime: "2020-07-26T12:32:56Z"
       message: All legacy kube types successfully parsed
       reason: ParsedGroupVersions
       status: "False"
       type: NamespaceDeletionGroupVersionParsingFailure
     - lastTransitionTime: "2020-07-26T12:32:56Z"
       message: All content successfully deleted, may be waiting on finalization
       reason: ContentDeleted
       status: "False"
       type: NamespaceDeletionContentFailure
     - lastTransitionTime: "2020-07-26T12:32:56Z"
       message: 'Some resources are remaining: cephobjectstoreusers.ceph.rook.io has
         1 resource instances'
       reason: SomeResourcesRemain
       status: "True"
       type: NamespaceContentRemaining
     - lastTransitionTime: "2020-07-26T12:32:56Z"
       message: 'Some content in the namespace has finalizers remaining: cephobjectstoreuser.ceph.rook.io
         in 1 resource instances'
       reason: SomeFinalizersRemain
       status: "True"
       type: NamespaceFinalizersRemaining
   ```

3. Delete all the remaining resources listed in the previous step.
   For each of the resources to be deleted, perform the following:

   a. Get the object kind of the resource which needs to be removed.
      See the message in the previous output.
      Example, where `cephobjectstoreuser.ceph.rook.io` is the object kind.

      ```
      message: Some content in the namespace has finalizers remaining:
      cephobjectstoreuser.ceph.rook.io
      ```

   b. Get the Object name corresponding to the object kind.

      ```
      oc get <Object-kind> -n <project-name>
      ```

      Example:

      ```
      oc get cephobjectstoreusers.ceph.rook.io -n openshift-storage
      ```

      Example output:

      ```
      NAME                          AGE
      noobaa-ceph-objectstore-user   26h
      ```

   c. Patch the resources.

      ```
      oc patch -n <project-name> <object-kind>/<object-name> --type=merge -p '{"metadata": {"finalizers":null}}'
      ```

Example :

```
oc patch -n openshift-storage cephobjectstoreusers.ceph.rook.io/noobaa-ceph-objectstore-user \
--type=merge -p '{"metadata": {"finalizers":null}}'
```

Example output:

```
cephobjectstoreuser.ceph.rook.io/noobaa-ceph-objectstore-user patched
```

4. Verify that the openshift-storage project is deleted.

```
oc get project openshift-storage
```

Example output:

```
Error from server (NotFound): namespaces "openshift-storage" not found
```

What to do next

If the issue persists, contact IBM Support.

Troubleshooting CephFS PVC creation in external mode

If you have updated the IBM Storage Ceph cluster from a version lower than 4.1.1 to the latest release and is not a freshly deployed cluster, you must manually set the application type for CephFS pool on the IBM Storage Ceph cluster to enable CephFS PVC creation in external mode.

Procedure

1. Check for CephFS pvc stuck in *Pending* status.

```
oc get pvc -n <namespace>
```

Example output :

```
NAME                         STATUS    VOLUME
CAPACITY   ACCESS MODES    STORAGECLASS                        AGE
ngx-fs-pxknkcix20-pod        Pending
                            ocs-external-storagecluster-cephfs   28h
[...]
```

2. Check the `describe` output to see the events for respective PVC.
Expected error message is:

```
cephfs_metadata/csi.volumes.default/csi.volume.pvc-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx: (1) Operation not permitted)
```

```
oc describe pvc ngx-fs-pxknkcix20-pod -n nginx-file
```

Example output:

```
Name:          ngx-fs-pxknkcix20-pod
Namespace:     nginx-file
StorageClass:  ocs-external-storagecluster-cephfs
Status:        Pending
Volume:
Labels:        <none>
Annotations:   volume.beta.kubernetes.io/storage-provisioner: openshift-storage.cephfs.csi.ceph.com
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:    Filesystem
Mounted By:    ngx-fs-oyoe047v2bn2ka42jfgg-pod-hqhzf
Events:
 Type     Reason              Age                    From
Message
 ----     ------              ----                   ----
-------
 Warning  ProvisioningFailed  107m (x245 over 22h)  openshift-storage.cephfs.csi.ceph.com_csi-cephfsplugin-provisioner-
5f8b66cc96-hvcqp_6b7044af-c904-4795-9ce5-bf0cf63cc4a4
 (combined from similar events): failed to provision volume with StorageClass "ocs-external-storagecluster-cephfs": rpc
error: code = Internal desc = error (an error (exit status 1) occurred while
 running rados args: [-m 192.168.13.212:6789,192.168.13.211:6789,192.168.13.213:6789 --id csi-cephfs-provisioner --
keyfile=stripped -c /etc/ceph/ceph.conf -p cephfs_metadata getomapval
 csi.volumes.default csi.volume.pvc-1ac0c6e6-9428-445d-bbd6-1284d54ddb47 /tmp/omap-get-186436239 --namespace=csi])
occurred, command output streams is ( error getting omap value
 cephfs_metadata/csi.volumes.default/csi.volume.pvc-1ac0c6e6-9428-445d-bbd6-1284d54ddb47: (1) Operation not permitted)
```

3. Check the settings for the *<cephfs metadata pool name>*.
In this example, *<cephfs metadata pool name>* is `cephfs_metadata` and *<cephfs data pool name>* is `cephfs_data`. For running the command, you need `jq` preinstalled in the IBM Storage Ceph client node.

```
ceph osd pool ls detail --format=json | jq '.[] | select(.pool_name| startswith("cephfs")) | .pool_name,
.application_metadata' "cephfs_data"
{
  "cephfs": {}
}
"cephfs_metadata"
{
```

```
   "cephfs": {}
}
```

4. Set the application type for CephFS pool.
   Run the following commands on the IBM Storage Ceph client node:

```
ceph osd pool application set <cephfs metadata pool name> cephfs metadata cephfs ceph osd pool application set <cephfs
data pool name> cephfs data cephfs
```

5. Verify if the settings are applied.

```
ceph osd pool ls detail --format=json | jq '.[] | select(.pool_name| startswith("cephfs")) | .pool_name,
.application_metadata' "cephfs_data"


{ "cephfs": { "data": "cephfs" } } "cephfs_metadata" { "cephfs": { "metadata": "cephfs" } }
```

6. Check the CephFS PVC status again.
   The PVC should now be in *Bound* state.

```
oc get pvc -n <namespace>
```

   Example output :

```
NAME                        STATUS    VOLUME
CAPACITY  ACCESS MODES    STORAGECLASS                          AGE
ngx-fs-pxknkcix20-pod     Bound     pvc-1ac0c6e6-9428-445d-bbd6-1284d54ddb47
1Mi       RWO             ocs-external-storagecluster-cephfs   29h
[...]
```

Restoring the monitor pods in Fusion Data Foundation

Use this information to manually restore the monitor pods Fusion Data Foundation, when necessary.

About this task

Restore the monitor pods if all three of them go down, and when Fusion Data Foundation is not able to recover the monitor pods automatically.

Note: This is a disaster recovery procedure and must be performed under the guidance of the IBM support team. Contact IBM support team.

Procedure

1. Scale down the `rook-ceph-operator` and `ocs-operator`.

```
oc scale deployment rook-ceph-operator --replicas=0 -n openshift-storage

oc scale deployment ocs-operator --replicas=0 -n openshift-storage
```

2. Create a backup of all deployments in `openshift-storage` namespace.

```
mkdir backup

cd backup

oc project openshift-storage

for d in $(oc get deployment|awk -F' ' '{print $1}'|grep -v NAME); do echo $d;oc get deployment $d -o yaml >
oc_get_deployment.${d}.yaml; done
```

3. Patch the MDS deployments to remove the livenessProbe parameter and run it with the command parameter as sleep.

```
for i in $(oc get deployment -l app=rook-ceph-osd -oname);do oc patch ${i} -n openshift-storage --type='json' -p
'[{"op":"remove", "path":"/spec/template/spec/containers/0/livenessProbe"}]' ; oc patch ${i} -n openshift-storage -p
'{"spec": {"template": {"spec": {"containers": [{"name": "osd", "command": ["sleep", "infinity"], "args": []}]}}}}' ; done
```

4. Retrieve the `monstore` cluster map from all the OSDs.
   a. Create the recover_mon.sh script.

```
!/bin/bash
ms=/tmp/monstore

rm -rf $ms
mkdir $ms

for osd_pod in $(oc get po -l app=rook-ceph-osd -oname -n openshift-storage); do

  echo "Starting with pod: $osd_pod"

  podname=$(echo $osd_pod|sed 's/pod\///g')
  oc exec $osd_pod -- rm -rf $ms
  oc cp $ms $podname:$ms

  rm -rf $ms
  mkdir $ms

  echo "pod in loop: $osd_pod ; done deleting local dirs"

  oc exec $osd_pod -- ceph-objectstore-tool --type bluestore --data-path /var/lib/ceph/osd/ceph-$(oc get $osd_pod -
```

```
ojsonpath='{ .metadata.labels.ceph_daemon_id }') --op update-mon-db --no-mon-config --mon-store-path $ms
  echo "Done with COT on pod: $osd_pod"

  oc cp $podname:$ms $ms

  echo "Finished pulling COT data from pod: $osd_pod"
done
```

b. Run the recover_mon.sh script.

```
chmod +x recover_mon.sh

./recover_mon.sh
```

5. Patch the MON deployments, and run it with the command parameter as sleep.
   a. Edit the MON deployments.

```
for i in $(oc get deployment -l app=rook-ceph-mon -oname);do oc patch ${i} -n openshift-storage -p '{"spec":
{"template": {"spec": {"containers": [{"name": "mon", "command": ["sleep", "infinity"], "args": []}]}}}}'; done
```

   b. Patch the MON deployments to increase the initialDelaySeconds.

```
oc get deployment rook-ceph-mon-a -o yaml | sed "s/initialDelaySeconds: 10/initialDelaySeconds: 2000/g" | oc replace
-f -

oc get deployment rook-ceph-mon-b -o yaml | sed "s/initialDelaySeconds: 10/initialDelaySeconds: 2000/g" | oc replace
-f -

oc get deployment rook-ceph-mon-c -o yaml | sed "s/initialDelaySeconds: 10/initialDelaySeconds: 2000/g" | oc replace
-f -
```

6. Copy the previously retrieved `monstore` to the **mon-a** pod.

```
oc cp /tmp/monstore/ $(oc get po -l app=rook-ceph-mon,mon=a -oname |sed 's/pod\///g'):/tmp/
```

7. Navigate into the MON pod and change the ownership of the retrieved `monstore`.

```
oc rsh $(oc get po -l app=rook-ceph-mon,mon=a -oname)

 chown -R ceph:ceph /tmp/monstore
```

8. Copy the keyring template file before rebuilding the mon db.

```
oc rsh $(oc get po -l app=rook-ceph-mon,mon=a -oname)

 cp /etc/ceph/keyring-store/keyring /tmp/keyring

 cat /tmp/keyring
  [mon.]
    key = AQCleqldWqm5IhAAgZQbEzoShkZV42RiQVffnA==
    caps mon = "allow *"
  [client.admin]
    key = AQCmAKld8J05KxAArOWeRAw63gAwwZO5o75ZNQ==
    auid = 0
    caps mds = "allow *"
    caps mgr = "allow *"
    caps mon = "allow *"
    caps osd = "allow *”
```

9. Identify the keyring of all other Ceph daemons (MGR, MDS, RGW, Crash, CSI and CSI provisioners) from its respective secrets.

```
oc get secret rook-ceph-mds-ocs-storagecluster-cephfilesystem-a-keyring -ojson  | jq .data.keyring | xargs echo | base64 -
d

[mds.ocs-storagecluster-cephfilesystem-a]
key = AQB3r8VgAtr6OhAAVhhXpNKqRTuEVdRoxG4uRA==
caps mon = "allow profile mds"
caps osd = "allow *"
caps mds = "allow"
```

Example keyring file, /etc/ceph/ceph.client.admin.keyring:

```
[mon.]
        key = AQDxTF1hNgLTNxAAi51cCojs01b4I5E6v2H8Uw==
        caps mon = "allow "
[client.admin]
        key = AQDxTF1hpzguOxAA0sS8nN4udoO35OEbt3bqMQ==
        caps mds = "allow "
        caps mgr = "allow *"
        caps mon = "allow *"
        caps osd = "allow *"
[mds.ocs-storagecluster-cephfilesystem-a]
        key = AQCKTV1horgjARAA8aF/BDh/4+eG4RCNBCl+aw==
        caps mds = "allow"
        caps mon = "allow profile mds"
        caps osd = "allow *"
[mds.ocs-storagecluster-cephfilesystem-b]
        key = AQCKTV1hN4gKLBAA5emIVq3ncV7AMEM1c1RmGA==
        caps mds = "allow"
        caps mon = "allow profile mds"
        caps osd = "allow *"
[client.rgw.ocs.storagecluster.cephobjectstore.a]
        key = AQCOkdBixmpiAxAA4X7zjn6SGTI9c1MBflszYA==
        caps mon = "allow rw"
        caps osd = "allow rwx"
[mgr.a]
```

```
        key = AQBOTV1hGYOEORAA87471+eIZLZtptfkcHvTRg==
        caps mds = "allow *"
        caps mon = "allow profile mgr"
        caps osd = "allow *"
[client.crash]
        key = AQBOTV1htO1aGRAAe2MPYcGdiAT+Oo4CNPSF1g==
        caps mgr = "allow rw"
        caps mon = "allow profile crash"
[client.csi-cephfs-node]
        key = AQBOTV1hiAtuBBAAaPPBVgh1AqZJlDeHWdoFLw==
        caps mds = "allow rw"
        caps mgr = "allow rw"
        caps mon = "allow r"
        caps osd = "allow rw tag cephfs *="
[client.csi-cephfs-provisioner]
        key = AQBNTV1hHu6wMBAAzNXZv36aZJuE1iz7S7GfeQ==
        caps mgr = "allow rw"
        caps mon = "allow r"
        caps osd = "allow rw tag cephfs metadata="
[client.csi-rbd-node]
        key = AQBNTV1h+LnkIRAAWnpIN9bUAmSHOvJ0EJXHRw==
        caps mgr = "allow rw"
        caps mon = "profile rbd"
        caps osd = "profile rbd"
[client.csi-rbd-provisioner]
        key = AQBNTV1hMNcsExAAvA3gHB2qaY33LOdWCvHG/A==
        caps mgr = "allow rw"
        caps mon = "profile rbd"
        caps osd = "profile rbd"
```

Important:

- For `client.csi` related keyring, refer to the previous keyring file output and add the default `caps` after fetching the key from its respective Fusion Data Foundation secret.
- OSD keyring is added automatically post recovery.

10. Navigate into the **mon-a** pod, and verify that the `monstore` has `monmap`.
    a. Navigate into the **mon-a** pod.

    ```
    oc rsh $(oc get po -l app=rook-ceph-mon,mon=a -oname)
    ```

    b. Verify that the `monstore` has `monmap`.

    ```
    ceph-monstore-tool /tmp/monstore get monmap -- --out /tmp/monmap
    ```

    ```
    monmaptool /tmp/monmap --print
    ```

11. Optional: If the `monmap` is missing then create a new `monmap`.

    ```
    monmaptool --create --add <mon-a-id><mon-a-ip> --add <mon-b-id><mon-b-ip> --add <mon-c-id><mon-c-ip> --enable-all-features --clobber /root/monmap --fsid <fsid>
    ```

    <mon-a-id>
    Is the ID of the **mon-a** pod.
    <mon-a-ip>
    Is the IP address of the **mon-a** pod.
    <mon-b-id>
    Is the ID of the **mon-b** pod.
    <mon-b-ip>
    Is the IP address of the **mon-b** pod.
    <mon-c-id>
    Is the ID of the **mon-c** pod.
    <mon-c-ip>
    Is the IP address of the **mon-c** pod.
    <fsid>
    Is the file system ID.

12. Verify the `monmap`.

    ```
    monmaptool /root/monmap --print
    ```

13. Import the `monmap`.
    Important: Use the previously created keyring file.

    ```
    ceph-monstore-tool /tmp/monstore rebuild -- --keyring /tmp/keyring --monmap /root/monmap
    ```

    ```
    chown -R ceph:ceph /tmp/monstore
    ```

14. Create a backup of the old `store.db` file.

    ```
    mv /var/lib/ceph/mon/ceph-a/store.db /var/lib/ceph/mon/ceph-a/store.db.corrupted
    ```

    ```
    mv /var/lib/ceph/mon/ceph-b/store.db /var/lib/ceph/mon/ceph-b/store.db.corrupted
    ```

    ```
    mv /var/lib/ceph/mon/ceph-c/store.db /var/lib/ceph/mon/ceph-c/store.db.corrupted
    ```

15. Copy the rebuild store.db file to the monstore directory.

    ```
    mv /tmp/monstore/store.db /var/lib/ceph/mon/ceph-a/store.db
    ```

    ```
    chown -R ceph:ceph /var/lib/ceph/mon/ceph-a/store.db
    ```

16. After rebuilding the monstore directory, copy the store.db file from local to the rest of the MON pods, where *<id>* is the ID of the MON pod.

    ```
    oc cp $(oc get po -l app=rook-ceph-mon,mon=a -oname | sed 's/pod\///g'):/var/lib/ceph/mon/ceph-a/store.db /tmp/store.db
    ```

```
oc cp /tmp/store.db $(oc get po -l app=rook-ceph-mon,mon=<id> -oname | sed 's/pod\///g'):/var/lib/ceph/mon/ceph-<id>
```

17. Navigate into the rest of the MON pods and change the ownership of the copied `monstore`, where *<id>* is the ID of the MON pod.

```
oc rsh $(oc get po -l app=rook-ceph-mon,mon=<id> -oname)
```

```
chown -R ceph:ceph /var/lib/ceph/mon/ceph-<id>/store.db
```

18. Revert the patched changes.
Important: Ensure that the MON, MGR, and OSD pods are up and running.

    For MON deployments
    Use the following command, where *<mon-deployment.yaml>* is the MON deployment yaml file.

    ```
    oc replace --force -f <mon-deployment.yaml>
    ```

    For OSD deployments:
    Use the following command, where *<osd-deployment.yaml>* is the OSD deployment yaml file.

    ```
    oc replace --force -f <osd-deployment.yaml>
    ```

    For MGR deployments
    Use the following command, where *<mgr-deployment.yaml>* is the MGR deployment yaml file.

    ```
    oc replace --force -f <mgr-deployment.yaml>
    ```

19. Scale up the `rook-ceph-operator` and `ocs-operator` deployments.

```
oc -n openshift-storage scale deployment ocs-operator --replicas=1
```

What to do next

1. Check the Ceph status to confirm that CephFS is running.

```
ceph -s
```

Example output:

```
cluster:
   id:     f111402f-84d1-4e06-9fdb-c27607676e55
   health: HEALTH_ERR
           1 filesystem is offline
           1 filesystem is online with fewer MDS than max_mds
           3 daemons have recently crashed

   services:
     mon: 3 daemons, quorum b,c,a (age 15m)
     mgr: a(active, since 14m)
     mds: ocs-storagecluster-cephfilesystem:0
     osd: 3 osds: 3 up (since 15m), 3 in (since 2h)

   data:
     pools:   3 pools, 96 pgs
     objects: 500 objects, 1.1 GiB
     usage:   5.5 GiB used, 295 GiB / 300 GiB avail
     pgs:     96 active+clean
```

2. Check the Multicloud Object Gateway (MCG) status. It should be active, and the backingstore and bucketclass should be in *Ready* state.

```
noobaa status -n openshift-storage
```

Note: If the MCG is not in the active state, and the backingstore and bucketclass not in the `Ready` state, you need to restart all the MCG related pods. For more information, see Restoring the Multicloud Object Gateway.

- **Restoring the Multicloud Object Gateway**
  Follow these steps to restore the Multicloud Object Gateway.

Restoring the Multicloud Object Gateway

Follow these steps to restore the Multicloud Object Gateway.

About this task

If the Multicloud Object Gateway (MCG) is not in the *Active* state, and the backingstore and bucketclass is not in the *Ready* state, you need to restart all the MCG related pods, and check the MCG status to confirm that the MCG is back up and running.

Procedure

1. Restart all the pods related to the MCG.

```
oc delete pods <noobaa-operator> -n openshift-storage
```

```
oc delete pods <noobaa-core> -n openshift-storage
```

```
oc delete pods <noobaa-endpoint> -n openshift-storage
```

```
oc delete pods <noobaa-db> -n openshift-storage
```

<noobaa-operator>
: Is the name of the MCG operator

<noobaa-core>
: Is the name of the MCG core pod

<noobaa-endpoint>
: Is the name of the MCG endpoint

<noobaa-db>
: Is the name of the MCG db pod

2. If the RADOS Object Gateway (RGW) is configured, restart the pod.

```
oc delete pods <rgw-pod> -n openshift-storage
```

<rgw-pod>
: Is the name of the RGW pod

Note: In OpenShift Container Platform 4.11, after the recovery, RBD PVC fails to get mounted on the application pods. Hence, you need to restart the node that is hosting the application pods. To get the node name that is hosting the application pod, run the following command:

```
oc get pods <application-pod> -n <namespace> -o yaml | grep nodeName
  nodeName: node_name
```

Restoring ceph-monitor quorum in Fusion Data Foundation

In some circumstances, the **ceph-mons** might lose quorum. If the **mons** cannot form quorum again, there is a manual procedure to get the quorum going again. The only requirement is that, at least one **mon** must be healthy. Use this information to remove the unhealthy **mons** from quorum and form a quorum again with a single **mon**, then bring the quorum back to the original size.

About this task

An example use case is if you have three **mons** and lose quorum, you need to remove the two bad **mons** from quorum, notify the good **mon** that it is the only **mon** in quorum, and then restart the good **mon**.

Procedure

1. Stop the **rook-ceph-operator** so that the **mons** are not failed over when you are modifying the **monmap**.

```
oc -n openshift-storage scale deployment rook-ceph-operator --replicas=0
```

2. Inject a new **monmap**.
Note: You must inject the **monmap** very carefully. If run incorrectly, your cluster could be permanently destroyed. The Ceph **monmap** keeps track of the **mon** quorum. The **monmap** is updated to only contain the healthy mon. In this example, the healthy mon is **rook-ceph-mon-b**, while the unhealthy **mons** are **rook-ceph-mon-a** and **rook-ceph-mon-c**.
   a. Take a backup of the current **rook-ceph-mon-b** deployment:

```
oc -n openshift-storage get deployment rook-ceph-mon-b -o yaml > rook-ceph-mon-b-deployment.yaml
```

   b. Open the YAML file and copy the command and args (arguments) from the **mon** container.
   These are part of the containers list, as shown in the following example. This is needed for the **monmap** changes.

```
[...]
  containers:
  - args:
    - --fsid=41a537f2-f282-428e-989f-a9e07be32e47
    - --keyring=/etc/ceph/keyring-store/keyring
    - --log-to-stderr=true
    - --err-to-stderr=true
    - --mon-cluster-log-to-stderr=true
    - '--log-stderr-prefix=debug '
    - --default-log-to-file=false
    - --default-mon-cluster-log-to-file=false
    - --mon-host=$(ROOK_CEPH_MON_HOST)
    - --mon-initial-members=$(ROOK_CEPH_MON_INITIAL_MEMBERS)
    - --id=b
    - --setuser=ceph
    - --setgroup=ceph
    - --foreground
    - --public-addr=10.100.13.242
    - --setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db
    - --public-bind-addr=$(ROOK_POD_IP)
    command:
    - ceph-mon
[...]
```

   c. Cleanup the copied command and args fields to form a pastable command as follows:

```
# ceph-mon \
    --fsid=41a537f2-f282-428e-989f-a9e07be32e47 \
    --keyring=/etc/ceph/keyring-store/keyring \
    --log-to-stderr=true \
    --err-to-stderr=true \
```

```
    --mon-cluster-log-to-stderr=true \
    --log-stderr-prefix=debug \
    --default-log-to-file=false \
    --default-mon-cluster-log-to-file=false \
    --mon-host=$ROOK_CEPH_MON_HOST \
    --mon-initial-members=$ROOK_CEPH_MON_INITIAL_MEMBERS \
    --id=b \
    --setuser=ceph \
    --setgroup=ceph \
    --foreground \
    --public-addr=10.100.13.242 \
    --setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db \
    --public-bind-addr=$ROOK_POD_IP
```

Note: Ensure to remove the single quotes around the `--log-stderr-prefix` flag and the parenthesis around the variables being passed `ROOK_CEPH_MON_HOST`, `ROOK_CEPH_MON_INITIAL_MEMBERS`, and `ROOK_POD_IP`).

d. Patch the `rook-ceph-mon-b` deployment to stop the working of this `mon` without deleting the `mon` pod.

```
oc -n openshift-storage patch deployment rook-ceph-mon-b  --type='json' -p '[{"op":"remove", "path":"/spec/template/spec/containers/0/livenessProbe"}]'
```

```
oc -n openshift-storage patch deployment rook-ceph-mon-b -p '{"spec": {"template": {"spec": {"containers": [{"name": "mon", "command": ["sleep", "infinity"], "args": []}]}}}}'
```

e. Perform the following steps on the `mon-b` pod:

i. Connect to the pod of a healthy `mon` and run the following command:

```
oc -n openshift-storage exec -it <mon-pod> bash
```

ii. Set the variable.

```
monmap_path=/tmp/monmap
```

iii. Extract the `monmap` to a file, by pasting the ceph `mon` command from the good `mon` deployment and adding the `--extract-monmap=${monmap_path}` flag.

```
# ceph-mon \
       --fsid=41a537f2-f282-428e-989f-a9e07be32e47 \
       --keyring=/etc/ceph/keyring-store/keyring \
       --log-to-stderr=true \
       --err-to-stderr=true \
       --mon-cluster-log-to-stderr=true \
       --log-stderr-prefix=debug \
       --default-log-to-file=false \
       --default-mon-cluster-log-to-file=false \
       --mon-host=$ROOK_CEPH_MON_HOST \
       --mon-initial-members=$ROOK_CEPH_MON_INITIAL_MEMBERS \
       --id=b \
       --setuser=ceph \
       --setgroup=ceph \
       --foreground \
       --public-addr=10.100.13.242 \
       --setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db \
       --public-bind-addr=$ROOK_POD_IP \
       --extract-monmap=${monmap_path}
```

iv. Review the contents of the `monmap`.

```
monmaptool --print /tmp/monmap
```

v. Remove the bad `mons` from the `monmap`.

```
monmaptool ${monmap_path} --rm <bad_mon>
```

In this example we remove `mon0` and `mon2`:

```
monmaptool ${monmap_path} --rm a
# monmaptool ${monmap_path} --rm c
```

vi. Inject the modified `monmap` into the good `mon`, by pasting the ceph `mon` command and adding the `--inject-monmap=${monmap_path}` flag as follows:

```
# ceph-mon \
       --fsid=41a537f2-f282-428e-989f-a9e07be32e47 \
       --keyring=/etc/ceph/keyring-store/keyring \
       --log-to-stderr=true \
       --err-to-stderr=true \
       --mon-cluster-log-to-stderr=true \
       --log-stderr-prefix=debug \
       --default-log-to-file=false \
       --default-mon-cluster-log-to-file=false \
       --mon-host=$ROOK_CEPH_MON_HOST \
       --mon-initial-members=$ROOK_CEPH_MON_INITIAL_MEMBERS \
       --id=b \
       --setuser=ceph \
       --setgroup=ceph \
       --foreground \
       --public-addr=10.100.13.242 \
       --setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db \
       --public-bind-addr=$ROOK_POD_IP \
       --inject-monmap=${monmap_path}
```

vii. Exit the shell to continue.

3. Edit the Rook `configmaps`.

a. Edit the `configmap` that the operator uses to track the `mons`.

```
oc -n openshift-storage edit configmap rook-ceph-mon-endpoints
```

b. Verify that in the data element you see three `mons` such as the following (or more depending on your `moncount`):

```
data: a=10.100.35.200:6789;b=10.100.13.242:6789;c=10.100.35.12:6789
```

c. Delete the bad `mons` from the list to end up with a single good `mon`. For example:

```
data: b=10.100.13.242:6789
```

d. Save the file and exit.

e. Adapt a `Secret` which is used for the `mons` and other components.

i. Set a value for the variable *good_mon_id*.

For example:

```
good_mon_id=b
```

ii. You can use the `oc patch` command to patch the `rook-ceph-config` secret and update the two key/value pairs `mon_host` and `mon_initial_members`.

```
mon_host=$(oc -n openshift-storage get svc rook-ceph-mon-b -o jsonpath='{.spec.clusterIP}')
oc -n openshift-storage patch secret rook-ceph-config -p '{"stringData": {"mon_host": "
[v2:'"${mon_host}"':3300,v1:'"${mon_host}"':6789]", "mon_initial_members": "'"${good_mon_id}"'"}}'
```

Note: If you are using `hostNetwork: true`, you need to replace the `mon_host` var with the node IP the `mon` is pinned to (`nodeSelector`). This is because there is no `rook-ceph-mon-*` service created in that "mode".

4. Restart the `mon`. You need to restart the good `mon` pod with the original `ceph-mon` command to pick up the changes.

a. Use the `oc replace` command on the backup of the `mon` deployment YAML file:

```
oc replace --force -f rook-ceph-mon-b-deployment.yaml
```

Note: Option `--force` deletes the deployment and creates a new one.

b. Verify the status of the cluster. The status should show one `mon` in quorum. If the status looks good, your cluster should be healthy again.

5. Delete the two mon deployments that are no longer expected to be in quorum.
For example:

```
oc delete deploy <rook-ceph-mon-1>
oc delete deploy <rook-ceph-mon-2>
```

In this example the deployments to be deleted are `rook-ceph-mon-a` and `rook-ceph-mon-c`.

6. Restart the operator.

a. Start the rook operator again to resume monitoring the health of the cluster.
Note: It is safe to ignore the errors that a number of resources already exist.

```
oc -n openshift-storage scale deployment rook-ceph-operator --replicas=1
```

The operator automatically adds more `mons` to increase the quorum size again depending on the `mon` count.

Changing resources for the Fusion Data Foundation components

When you install Fusion Data Foundation, it comes with pre-defined resources that the Fusion Data Foundation pods can consume. In some situations with higher I/O load, it might be required to increase these limits.

- To change the CPU and memory resources on the rook-ceph pods, see Changing the CPU and memory resources on the rook-ceph pods.
- To tune the resources for the Multicloud Object Gateway (MCG), see Tuning the resources for the MCG.

- **Changing the CPU and memory resources on the rook-ceph pods**
When you install Fusion Data Foundation, it comes with pre-defined CPU and memory resources for the rook-ceph pods. You can manually increase these values according to the requirements.
- **Tuning the resources for the MCG**
The default configuration for the Multicloud Object Gateway (MCG) is optimized for low resource consumption and not performance.

Changing the CPU and memory resources on the rook-ceph pods

When you install Fusion Data Foundation, it comes with pre-defined CPU and memory resources for the rook-ceph pods. You can manually increase these values according to the requirements.

About this task

You can change the CPU and memory resources on the following pods:

- mgr
- mds
- rgw

The following example illustrates how to change the CPU and memory resources on the rook-ceph pods. In this example, the existing MDS pod values of `cpu` and `memory` are increased from `1` and `4Gi` to `2` and `8Gi` respectively.

Procedure

1. Edit the storage cluster.
   Use the following command, where *<storagecluster_name>* specifies the name of the storage cluster.

   ```
   oc edit storagecluster -n openshift-storage <storagecluster_name>
   ```

   For example:

   ```
   oc edit storagecluster -n openshift-storage ocs-storagecluster
   ```

2. Add the following lines to the storage cluster Custom Resource (CR):

   ```
   spec:
     resources:
       mds:
         limits:
           cpu: 2memory: 8Gi
         requests:
           cpu: 2memory: 8Gi
   ```

3. Save the changes and exit the editor.
4. Alternatively, run the **oc patch** command to change the CPU and memory value of the `mds` pod.
   Use the following command, where *<storagecluster_name>* specifies the name of the storage cluster.

   ```
   oc patch -n openshift-storage storagecluster <storagecluster_name> --type merge \ --patch '{"spec": {"resources": {"mds": {"limits": {"cpu": "2","memory": "8Gi"},"requests": {"cpu": "2","memory": "8Gi"}}}}}'
   ```

   For example:

   ```
   oc patch -n openshift-storage storagecluster ocs-storagecluster \
       --type merge \
       --patch '{"spec": {"resources": {"mds": {"limits": {"cpu": "2","memory": "8Gi"},"requests": {"cpu": "2","memory": "8Gi"}}}}}'
   ```

Tuning the resources for the MCG

The default configuration for the Multicloud Object Gateway (MCG) is optimized for low resource consumption and not performance.

For more information on how to tune the resources for the MCG, see the Red Hat Knowledgebase solution Performance tuning guide for Multicloud Object Gateway (NooBaa).

Disabling Multicloud Object Gateway external service after deploying OpenShift Data Foundation

When you deploy Fusion Data Foundation, public IPs are created even when OpenShift is installed as a private cluster. However, you can disable the Multicloud Object Gateway (MCG) load balancer usage by using the **disableLoadBalancerService** variable in the storagecluster CRD. This restricts MCG from creating any public resources for private clusters and helps to disable the NooBaa service **EXTERNAL-IP**.

Procedure

Run the following command and add the **disableLoadBalancerService** varialbe in the storagecluster YAML to set the service to ClusterIP:

```
oc edit storagecluster -n openshift-storage <storagecluster_name>
[...]
spec:
  arbiter: {}
  encryption:
    kms: {}
  externalStorage: {}
  managedResources:
    cephBlockPools: {}
    cephCluster: {}
    cephConfig: {}
    cephDashboard: {}
    cephFilesystems: {}
    cephNonResilientPools: {}
    cephObjectStoreUsers: {}
    cephObjectStores: {}
    cephRBDMirror: {}
    cephToolbox: {}
  mirroring: {}
  multiCloudGateway:
    disableLoadBalancerService: true       <--------------- Add this
    endpoints:
[...]
```

Note: To undo the changes and set the service to LoadBalancer, set the **disableLoadBalancerService** variable to **false** or remove that line completely.

Accessing `odf-console` with the `ovs-multitenant` plug-in by manually enabling global pod networking

If the **`odf-console`** plug-in fails with a timeout error when using the **`ovs-multitenant`** SDN plug-in, make the **`openshift-storage`** project's pod networking global to allow cross-namespace console communication.

In OpenShift Container Platform, when **`ovs-multitenant`**plug-in is used for software-defined networking (SDN), pods from different projects cannot send packets to or receive packets from pods and services of a different project. By default, pods cannot communicate between namespaces or projects because a project's pod networking is not global.

To access odf-console, the Red Hat OpenShift console pod in the **`openshift-console`** namespace needs to connect with the Fusion Data Foundation odf-console in the **`openshift-storage`** namespace. This is possible only when you manually enable global pod networking.

Issue

- When **`ovs-multitenant`** plug-in is used in the OpenShift Container Platform, the **`odf-console`** plug-in fails with the following message:

  ```
  GET request for "odf-console" plugin failed: Get "https://odf-console-service.openshift-storage.svc.cluster.local:9001/locales/en/plugin__odf-console.json": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
  ```

Resolution

- Make the pod networking for the Fusion Data Foundation project global:

  ```
  oc adm pod-network make-projects-global openshift-storage
  ```

Annotating encrypted RBD storage classes

This section provides you with information on annotating encrypted RBD storage classes.

When the OpenShift console creates a RADOS block device (RBD) storage class with encryption enabled, the annotation is set automatically. However, you need to add the annotation, **`cdi.kubevirt.io/clone-strategy=copy`** for any of the encrypted RBD storage classes that were previously created before updating to the Fusion Data Foundation. This enables customer data integration (CDI) to use host-assisted cloning instead of the default smart cloning.

The keys used to access an encrypted volume are tied to the namespace where the volume was created. When cloning an encrypted volume to a new namespace, such as, provisioning a new OpenShift Virtualization virtual machine, a new volume must be created and the content of the source volume must then be copied into the new volume. This behavior is triggered automatically if the storage class is properly annotated.

Force deletion of storage in provider clusters

Force-delete a storage consumer from the provider cluster when a client cluster was removed without completing the off-boarding process, to release storage space claimed by the deleted client.

Before you begin

- Administrator access to the OpenShift Data Foundation storage cluster in provider mode.

About this task

When a client cluster is deleted without performing the off-boarding process to remove all the resources from the corresponding provider cluster, you need to perform force deletion of the corresponding storage consumer from the provider cluster. This helps to release the storage space that was claimed by the client.

Procedure

1. Click **Storage** → **Storage Clients** from the OpenShift console.
2. Click the delete icon at the far right of the listed storage client cluster.
   The delete icon is enabled only after 5 minutes of the last heartbeat of the cluster.
3. Click **Confirm**.

Configuring CPU and memory resources for FDF-managed pods

You can define and adjust resource requests or limits for the pods managed by Fusion Data Foundation (FDF) to ensure optimal scheduling and performance of FDF workloads.

Configuring resources for operator pods

You can configure resources for operator pods, such as, **ocs-operator**, **odf-operator**, **noobaa-operator**, **ocs-client-operator**, and so on. You can edit the default resources in their Subscription as follows:

```
kind: Subscription
metadata:
  name: my-operator
spec:
  package: my-operator-package
  channel: alpha
  config:
    resources:
      requests:
        memory: <value>
        cpu: <value>
      limits:
        memory: <value>
        cpu: <value>
```

Configuring resources for pods deployed by `ocs-operator`

You can configure sub-resource pods deployed by **ocs-operator**, such as **rook-ceph-mgr**, **rook-ceph-mon**, **ocs-metrics-exporter** and so on. You can update the values for these pods by editing the resources field in the storage cluster as follows:

```
spec:
  resources:
    ocs-metrics-exporter:
      requests:
        cpu: <value>
        memory: <value>
      limits:
        cpu: <value>
        memory: <value>
```

Configuring CSI pods managed by `ceph-csi-operator`

You can configure CSI pods in the **openshift-storage** namespace that are managed by the **ceph-csi-operator**. You can edit the **drivers.csi.ceph.io** CR for each driver in the **openshift-storage** namespace and update the resource values for **ControllerPlugin/NodePlugin** resources based on the requirement as follows:

```
spec:
  controllerPlugin:
    resources:
      snapshotter:
        requests:
          cpu: <value>
          memory: <value>
        limits:
          cpu: <value>
          memory: <value>
```

Unable to delete incorrect IBM Scale (CNSA) cluster after first-time configuration

When incorrect or mock data is entered during first-time IBM Scale (CNSA) configuration with Fusion Data Foundation, a **Cluster** is created and recognized as Storage Area Network (SAN), cannot be deleted, and prevents a valid Scale connection.

Symptom

When you attempt to connect an IBM Scale (CNSA) cluster for the first time and provide incorrect or mock configuration data, the system does not create the expected **RemoteCluster** resource. Instead, a **Cluster** resource is created and is incorrectly identified as a SAN system in the Red Hat® OpenShift® Container Platform web console.

Because the Red Hat OpenShift Container Platform web console interprets the **Cluster** as an active SAN connection, the option to connect an IBM Scale cluster becomes disabled. You cannot delete the **Cluster** from the Red Hat OpenShift Container Platform web console or by using the CLI.

This issue prevents you from establishing a valid Scale cluster connection.

Cause

During the first-time IBM Scale cluster connection workflow, entering incorrect or mock configuration details results in incomplete resource creation. The **RemoteCluster** object fails to deploy, but the **Cluster** object is still created.

Because the platform assumes the **Cluster** corresponds to a SAN connection, the Red Hat OpenShift Container Platform web console automatically re-creates it after deletion attempts. As a result:

- The **Cluster** cannot be removed from the Red Hat OpenShift Container Platform web console.
- The **Cluster** reappears after CLI deletion attempts.
- The connection flow for IBM Scale becomes disabled because the system believes that a SAN connection is already established.

Resolution

To remove the incorrect **Cluster** entry and restore the ability to connect an IBM Scale cluster, do the following steps:

1. Delete the Scale cluster file system custom resource.

   Command example:

   ```
   oc delete filesystems.scale.spectrum.ibm.com SCALE_CONNECTION_NAME-FILESYSTEM_NAME
   ```

2. Delete the secret that contains the Scale cluster user details.

   Command example:

   ```
   oc delete secret SCALE_CONNECTION_NAME-user-details-secret
   ```

3. Delete the Scale cluster custom resource that was partially created.

   Command example:

   ```
   oc delete clusters.scale.spectrum.ibm.com ibm-spectrum-scale
   ```

4. Wait for the resource reconciliation cycle to complete.
5. Verify the following:

   a. Verify that the secret is deleted.

      Command example:

      ```
      oc get secret | grep SCALE_CONNECTION_NAME
      ```

   b. Verify that the file system custom resource is deleted.

      Command example:

      ```
      oc get filesystems.scale.spectrum.ibm.com
      ```

   c. Verify that the `Cluster` resource is no longer present in the `ibm-spectrum-fusion-ns` namespace.

      Command example:

      ```
      oc get cluster -n ibm-spectrum-fusion-ns
      ```

6. Go to the Red Hat OpenShift Container Platform web console and confirm that the Connect external systems > IBM Scale option is enabled again.
7. Reattempt the IBM Scale cluster connection by using the correct configuration details. For instructions, see Configuring with IBM Storage Scale.

Result

The incorrect `Cluster` entry is removed. The Red Hat OpenShift Container Platform web console re-enables the IBM Scale connection option, and you can establish a valid IBM Scale connection by using the correct details.

Filesystem deletion does not remove internal IBM Storage Scale resources

Deleting a filesystem custom resource does not automatically remove internal IBM Storage Scale remote cluster definitions.

Problem description

Deleting a filesystem custom resource (CR) in Red Hat® OpenShift® Container Platform does not fully clean up internal IBM Storage Scale resources.

When a filesystem CR is deleted, the IBM Storage Scale operator does not remove the underlying GPFS remote cluster definitions. As a result, the following GPFS resources remain present within the GPFS daemon pods on the worker nodes:

- `mmremotecluster`
- `mmremotefs`

Symptoms

After deleting the filesystem CR, you might experience the following behavior:

- Attempting to re-create the same filesystem CR fails.
- The filesystem CR does not reach the `Established` state.
- Errors occur due to existing internal GPFS remote cluster definitions.

Cause

The IBM Storage Scale operator does not automatically clean up low-level GPFS remote cluster definitions when a filesystem CR is deleted. These definitions persist within the GPFS daemon pods and block recreation of the filesystem.

Resolution

To remove the remaining internal GPFS remote cluster definitions, complete the following steps:

1. Log in to any IBM Storage Scale core pod.
2. List the available filesystems by running the following command:

`/usr/lpp/mmfs/bin/mmremotefs show`

3. Unmount the filesystem from all nodes by running the following command:

   `/usr/lpp/mmfs/bin/mmunmount scale_filesystem_name -a`

   Replace *scale_filesystem_name* with the name of your IBM Storage Scale filesystem.

4. Delete the filesystem by running the following command:

   `/usr/lpp/mmfs/bin/mmremotefs delete scale_filesystem_name`

   Replace *scale_filesystem_name* with the name of your IBM Storage Scale filesystem.

After the internal GPFS remote cluster definitions are removed, you can re-create the filesystem CR.

Result

Internal IBM Storage Scale remote cluster definitions are removed, and the filesystem can be successfully re-created.

Fusion Data Foundation service error scenarios

Use these troubleshooting information to know the problem and workaround when install or configure Fusion Data Foundation service.

- Fusion Data Foundation storage node failure
- Disk replacement and OpenShift dedicated OSD failure recovery
- Storage client status reporter pod is stuck in terminating state
- Fusion Data Foundation is in a degraded state
- Admission webhook warning
- Local storage operator unable to find candidate storage nodes
- Fusion Data Foundation capacity cannot be loaded
- Fusion Data Foundation cluster fails due to pending StorageClusterPreparing stage
- Data Foundation service configuration is incomplete
- Deletion of StorageCluster doesn't delete connected StorageClient
- The StorageClient pod is in ContainerStatusUnknown and CrashLoopBackOff state

Fusion Data Foundation is in a degraded state

Problem statement
: The Fusion Data Foundation storage cluster is in a degraded state.

Cause
: This issue might occurs due to many reasons. Follow the steps to find the root cause few the issues:

1. Log in to the OpenShift Container Platform console.
2. Go to Workloads > Pods.
3. Select the Openshift-storage form the Projects drop down.
4. Click CephClusters CR instance and go to YAML tab.
5. Check status field to find the exact failure reason. If the issue occurs because of MDS trimming, then follow steps mentioned in resolution section to resolve the issue.

Resolution

1. Log in to the OpenShift Container Platform console.
2. Go to Workloads > Pods.
3. Select the Openshift-storage form the Projects drop down.
4. Select the rook-ceph-operator pod from the list.
5. Go to Terminal tab, and run the following commands:

```
export CEPH_ARGS='-c /var/lib/rook/openshift-storage/openshift-storage.config'
ceph -s
```

Example output:

```
cluster:
    id:     0e7a1d51-ad98-4b33-ab71-4577de1b3e7d
    health: HEALTH_WARN
            2 MDSs behind on trimming

  services:
    mon: 3 daemons, quorum e,f,g (age 5d)
    mgr: a(active, since 14h), standbys: b
    mds: 1/1 daemons up, 1 hot standby
    osd: 5 osds: 5 up (since 6d), 5 in (since 6d)
    rgw: 1 daemon active (1 hosts, 1 zones)

  data:
    volumes: 1/1 healthy
    pools:   12 pools, 265 pgs
    objects: 1.54M objects, 1.3 TiB
    usage:   4.3 TiB used, 31 TiB / 35 TiB avail
    pgs:     265 active+clean
```

```
  io:
    client:   121 MiB/s rd, 119 MiB/s wr, 125 op/s rd, 165 op/s wr
```

```
ceph config get mds.{id} mds_log_max_segments 128
ceph config set mds mds_log_max_segments 256
ceph config get mds.{id} mds_log_max_segments 256
ceph -s
```

Example output:

```
cluster:
    id:     0e7a1d51-ad98-4b33-ab71-4577de1b3e7d
    health: HEALTH_OK

  services:
    mon: 3 daemons, quorum e,f,g (age 5d)
    mgr: a(active, since 14h), standbys: b
    mds: 1/1 daemons up, 1 hot standby
    osd: 5 osds: 5 up (since 6d), 5 in (since 6d)
    rgw: 1 daemon active (1 hosts, 1 zones)

  data:
    volumes: 1/1 healthy
    pools:   12 pools, 265 pgs
    objects: 1.54M objects, 1.3 TiB
    usage:   4.3 TiB used, 31 TiB / 35 TiB avail
    pgs:     265 active+clean

  io:
    client:   101 MiB/s rd, 109 MiB/s wr, 80 op/s rd, 128 op/s wr
```

For more information about the issue, see the MDSs behind on trimming in OCS4 Red Hat knowledgebase article.

Admission webhook warning

After you create Object Bucket Claim, the following error occurs:

```
Admission webhook warning :- ObjectBucketClaim my-obj-bucket violates policy 299 - "unknown field "spec.ss||"' after object bucket
```

For more information about this known issue, see Red Hat Known issues.

Local storage operator unable to find candidate storage nodes

Problem statement

When you configure a Fusion Data Foundation cluster, you do not find any candidate storage nodes.

Cause

When you configure Fusion Data Foundation cluster, only compute nodes with available disks (SSD/NVMe or HDD) get displayed in the Data Foundation page of IBM Fusion user interface. The following nodes get filtered out and do not display on the screen:

- Nodes have SSD/NVMe or HDD disks but they are not in available state
- The selected disk properties are not present in current node. For example, disk size or disk type.
- The total disk count (with same disk size, disk type) is less than 3.

Steps to verify whether you have the correct storage node candidates

1. In Red Hat OpenShift Container Platform console, go to Operators > Installed Operators.
2. Verify whether the `LocalStorage` operator is installed successfully.
3. Run the following command to get all the worker nodes:

   ```
   oc get node -l node-role.kubernetes.io/worker=
   ```

4. Run the following command to check if discovery results are created for all worker nodes.

   ```
   oc get localvolumediscoveryresult -n openshift-local-storage
   ```

5. Run the following command to confirm that none of the nodes have a Fusion Data Foundation storage label:

   ```
   oc get node -l cluster.ocs.openshift.io/openshift-storage=
   ```

Note: In Linux on IBM Z platform, disks might be formatted and partitioned first. For more information about this behavior, see Red Hat OpenShift Data Foundation on IBM Z and IBM LinuxONE - Reference Architecture section *4.1.1 and 4.1.2*.

If all the preceding checks pass, but the node still could not be seen in the IBM Fusion user interface, then contact IBM support.

Fusion Data Foundation capacity cannot be loaded

If you encounter this issue in the Data foundation page of IBM Fusion user interface, then contact IBM support.

Fusion Data Foundation cluster fails due to pending StorageClusterPreparing stage

Problem statement

Here, the PVC is not created and the `odfcluster` status shows as follows:

```
conditions:
  - lastTransitionTime: "2022-12-01T15:09:47Z"
```

```
    message: storagecluster is not ready,install pending
    reason: StorageClusterPreparing
    status: "False"
    type: Ready
  phase: InProgress
  replica: 1
```

Diagnosis and resolution

To diagnose and fix the problem, do the following steps:

1. Run the following command to open the `storagecluster` CR:

   ```
   oc get storageclusters.ocs.openshift.io -n openshift-storage ocs-storagecluster -o yaml
   ```

2. Check whether the output of the command shows the following error message in the status:

   ```
   ConfigMap "ocs-kms-connection-details" not found'
   ```

   Output example:

   ```
   status:
   conditions:
   - lastHeartbeatTime: "2023-03-29T08:01:10Z"
   lastTransitionTime: "2023-03-29T07:49:47Z"
   message: 'Error while reconciling: some StorageClasses were skipped while waiting
   for pre-requisites to be met: [ocs-storagecluster-cephfs,ocs-storagecluster-ceph-rbd]'
   reason: ReconcileFailed
   status: "False"
   type: ReconcileComplete
   ```

3. If you notice the error message, check the root-operator logs with the following command:

   ```
   oc logs -n openshift-storage $(oc get pod -n openshift-storage -l app=rook-ceph-operator -o name)
   ```

   Example output:

   ```
   2023-03-29 07:55:41.297073 E | ceph-cluster-controller: failed  to reconcile CephCluster  "openshift-storage/ocs-storagec
   cluster "ocs-storagecluster-cephcluster": failed to configure local ceph  cluster: failed to perform validation before cl
   request.
   URL: GET https://9.9.9.75:8200/v1/sys/mounts
   Code: 403. Errors: * permission denied
   ```

Data Foundation service configuration is incomplete

Problem statement

The Data Foundation storage cluster health is in a warning state.

Diagnosis and resolution

To diagnose and fix the problem, do the following steps:

1. Run the following command to check whether any alerts exist in the CEPH cluster.

   ```
   oc rsh -n openshift-storage $(oc get pods -n openshift-storage -o name -l app=rook-ceph-operator)
   ```

2. Run the following commands in the pod.

   ```
   sh-5.1$  export CEPH_ARGS='-c /var/lib/rook/openshift-storage/openshift-storage.config'

   sh-5.1$ ceph health detail
   ```

   Example output:

   ```
   HEALTH_WARN 3 daemons have recently crashed
   [WRN] RECENT_CRASH: 3 daemons have recently crashed
       osd.0 crashed on host rook-ceph-osd-prepare-ecf00ad4a684ca39c77ec263403cc29c-7vbn5 at 2024-07-29T09:04:58.992261Z
       osd.1 crashed on host rook-ceph-osd-prepare-f303dcfea8247471c14b744745e1b523-v1489 at 2024-07-29T09:04:59.283451Z
       osd.2 crashed on host rook-ceph-osd-prepare-a09bfae66e8278d3662bf8bb476a51fa-qzdln at 2024-07-29T09:04:59.932438Z
   ```

   ```
   sh-5.1$ ceph crash ls
   ```

   Example output:

   ```
   ID                                                                ENTITY  NEW
   2024-07-29T09:04:58.992261Z_0e656801-e235-44f5-b64f-939d81ba2dc1  osd.0    *
   2024-07-29T09:04:59.283451Z_b35f02e5-305b-42c4-bcbe-85fd5c0012fc  osd.1    *
   2024-07-29T09:04:59.932438Z_3618dc07-7756-405d-838e-7b627d73e64c  osd.2    *
   ```

3. Collect the DF must-gather to get more details about the issue. For steps to collect logs, see Collecting logs in IBM Fusion.
4. If you find any alerts in the CEPH cluster, then run the following command to clear the alerts.

   ```
   sh-5.1$ ceph crash prune 0
   ```

Deletion of `StorageCluster` doesn't delete connected `StorageClient`

Workaround

Remove the finalizer from the **`StorageClient`** and ensure that the **`StorageClient`** instance is not present on the cleaned-up cluster.

Note: This issue occurs on a reinstalled or reprovisioned cluster.

The `StorageClient` pod is in `ContainerStatusUnknown` and `CrashLoopBackOff` state

Diagnosis

When Fusion Data Foundation is reinstalled and reconfigured, the `StorageClient` continues pointing to the old instance. As a result, no or few storage classes are created, and NooBaa DB PVCs remain in the pending state.

Workaround

Check the metadata of the `StorageClient` CR. It contains details from the last connection. Force delete the old StorageClient instance and wait for it to be recreated.

Note: This issue occurs on a reinstalled or reprovisioned cluster.

- **Fusion Data Foundation storage node failure**
  You can do a node replacement proactively for an operational node and reactively for a failed node. For a failed node backed by local storage devices, you must replace the Fusion Data Foundation storage node.
- **Disk replacement and OpenShift dedicated OSD failure recovery**
  Learn how to replace a faulty disk in a Red Hat OpenShift dedicated Object Storage Device (OSD) cluster.
- **Storage client status reporter pod is stuck in terminating state**
  Learn how to diagnose and fix issues causing the storage client status reporter pod to get stuck during termination.

Fusion Data Foundation storage node failure

You can do a node replacement proactively for an operational node and reactively for a failed node. For a failed node backed by local storage devices, you must replace the Fusion Data Foundation storage node.

Before you begin

IBM recommends that replacement nodes be configured with similar infrastructure, resources, and disks as the node planned for replacement.

Note: Contact IBM Support before you proceed with any of these fixes.

Procedure

Do the following steps to check for the occurrence of Fusion Data Foundation storage node failure and identify the failed node:

1. Set the Fusion Data Foundation cluster to maintenance mode:

   ```
   oc label odfclusters.odf.isf.ibm.com -n ibm-spectrum-fusion-ns odfcluster "odf.isf.ibm.com/maintenanceMode=true"
   ```

   Example output:

   ```
   [root@fu40 ~]# oc label odfclusters.odf.isf.ibm.com -n ibm-spectrum-fusion-ns odfcluster
   "odf.isf.ibm.com/maintenanceMode=true"
   odfcluster.odf.isf.ibm.com/odfcluster labeled
   ```

2. Identify the failed node:
   a. Log in to IBM Fusion user interface.
   b. Go to Data foundation page and check for warning in the Health section for storage cluster.

   Alternatively, you can use the **oc** command to identify the node:

   ```
   oc get node -l cluster.ocs.openshift.io/openshift-storage=
   ```

   Sample output:

   ```
   [root@fu71-f09-vm3 ~]# oc get node -l cluster.ocs.openshift.io/openshift-storage=
   NAME                                STATUS     ROLES    AGE   VERSION
   f09-prc4m-worker-cluster-b-9chb5    NotReady   worker   27d   v1.24.0+4f0dd4d
   f09-prc4m-worker-cluster-c-mfb77    Ready      worker   31d   v1.24.0+4f0dd4d
   f09-prc4m-worker-cluster-d-r5bxx    Ready      worker   27d   v1.24.0+4f0dd4d
   ```

3. Identify the failed mon (if any) and Red Hat OpenShift Dedicated pods that are running in the node, which is planned for replacement:
   In an operational storage node environment:

   ```
   oc get pods -n openshift-storage -o wide | grep -i <node_name>
   ```

4. If the storage node failed in a failed storage node, there is no `node_name` for the failed pods. Filter the `pending` pods instead.

   ```
   oc get pods -n openshift-storage -o wide | grep -i pending
   ```

   Example output: The mon deployment is `rook-ceph-mon-d`, and the Red Hat OpenShift Dedicated deployment is `ook-ceph-osd-0`.

   ```
   [root@fu71-f09-vm3 ~]# oc get pods -n openshift-storage -o wide | grep -i pending
    rook-ceph-mon-d-67686857d7-zv62c                                  0/2     Pending     0          8m50s   <none>
   <none>                              <none>           <none>
    rook-ceph-osd-0-75b954c9bf-62xm4                                  0/2     Pending     0          8m50s   <none>
   <none>                              <none>           <none>
   ```

5. Remove the failed objects.

   Remove the failed node from `odfcluster` CR

   ```
   spec:
     autoScaleUp: false
     creator: CreatedByFusion
     deviceSets:
     - capacity: "0"
       count: 3
       name: ocs-deviceset-ibm-spectrum-fusion-local
       storageClass: ibm-spectrum-fusion-local
     encryption:
   ```

```
    keyManagementService: {}
  localVolumeSetSpec:
    deviceTypes:
    - disk
    - part
    size: 2Ti
  storageNodes:
  - f09-prc4m-worker-cluster-d-r5bxx
  - f09-prc4m-worker-cluster-c-mfb77
  - f09-prc4m-worker-cluster-b-9chb5 <<-- this one, remove this line.
```

Remove the mon and Red Hat OpenShift Dedicated pods

Scale down the deployments of the identified pods. The mon deployment is **rook-ceph-mon-d** and the Red Hat OpenShift Dedicated deployment is **rook-ceph-osd-0**.

```
oc scale deployment rook-ceph-mon-d --replicas=0 -n openshift-storage
oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

Ensure that you confirm the values of **mon_id** and **osd_id**.

Remove the crashcollector pods

Remove the crashcollector pods (if any). You must put scale replica to 0.

```
oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>  --replicas=0 -n openshift-
storage
```

Mark the failed node as unschedulable

Mark the node as **SchedulingDisabled**.

```
oc adm cordon <node_name>
```

Example command and output:

```
oc adm cordon f09-prc4m-worker-cluster-b-9chb5
node/f09-prc4m-worker-cluster-b-9chb5 cordoned
```

```
oc get node -l cluster.ocs.openshift.io/openshift-storage=
```

```
NAME                                STATUS                        ROLES    AGE   VERSION
f09-prc4m-worker-cluster-b-9chb5   NotReady,SchedulingDisabled   worker   28d   v1.24.0+4f0dd4d
f09-prc4m-worker-cluster-c-mfb77   Ready                         worker   31d   v1.24.0+4f0dd4d
f09-prc4m-worker-cluster-d-r5bxx   Ready                         worker   27d   v1.24.0+4f0dd4d
```

Remove the pods which are in Terminating state

This step is for the failed storage node. You can ignore this step if your are removing an operational node.

```
oc get pods -A -o wide | grep -i <node_name> |  awk '{if ($4 == "Terminating") system ("oc -n " $1 " delete pods
" $2  " --grace-period=0 " " --force ")}'
```

Example command and output:

```
oc get pods -A -o wide | grep -i f09-prc4m-worker-cluster-b-9chb5 |  awk '{if ($4 == "Terminating") system ("oc
-n " $1 " delete pods " $2  " --grace-period=0 " " --force ")}'
```

```
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resour
pod "isf-data-protection-operator-controller-manager-5c7cf574d5ms4xx" force deleted
```

Drain the node

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

Example command and output:

```
oc adm drain f09-prc4m-worker-cluster-b-9chb5 --force --delete-emptydir-data=true --ignore-daemonsets
node/f09-prc4m-worker-cluster-b-9chb5 already cordoned
```

```
WARNING: ignoring DaemonSet-managed Pods: openshift-cluster-csi-drivers/vmware-vsphere-csi-driver-node-7696f, opensh
node/f09-prc4m-worker-cluster-b-9chb5 drained
```

Delete the node

Delete the failed node:

```
oc delete node <node_name>
```

If you do not want to destroy this node for test purpose, you can remove the storage label **cluster.ocs.openshift.io/openshift-storage=**.

```
oc label nodes/f09-prc4m-worker-cluster-b-9chb5 cluster.ocs.openshift.io/openshift-storage-
```

6. Add new OpenShift® compute nodes.
   a. Create a new compute node and ensure that the new node is in ready state.
   b. Update new node info in **odfcluster** CR.

   Edit the odfcluster cr and add new node name in **StorageNodes**

   ```
   oc edit odfclusters.odf.isf.ibm.com -n ibm-spectrum-fusion-ns odfcluster
   ```

   ```
   storageNodes:
       - f09-prc4m-worker-cluster-b-djplp  <<--- new node
       - f09-prc4m-worker-cluster-d-r5bxx
       - f09-prc4m-worker-cluster-c-mfb77
   ```

   Verify whether the new nodes are labeled successfully as storage node.

   ```
   oc get node -l cluster.ocs.openshift.io/openshift-storage
   ```

```
NAME                                STATUS   ROLES    AGE   VERSION
f09-prc4m-worker-cluster-b-djplp   Ready    worker   27d   v1.24.0+4f0dd4d <<-- this node
f09-prc4m-worker-cluster-c-mfb77   Ready    worker   31d   v1.24.0+4f0dd4d
f09-prc4m-worker-cluster-d-r5bxx   Ready    worker   27d   v1.24.0+4f0dd4d
```

Verify whether the new PVs are created automatically. The local PV gets created automatically in a short time.

```
oc get pv | grep Available
```

```
local-pv-e97b23d7                          2Ti        RWO            Delete           Available
```

c. Replace the failed Red Hat OpenShift Dedicated disks.
Remove the failed Red Hat OpenShift Dedicated from the cluster. You can also specify multiple failed ODs. Use the correct **failed_osd_id**.

The **failed_osd_id** is the integer in the pod name immediately after the **rook-ceph-osd** prefix. You can add comma separated Red Hat OpenShift Dedicated IDs in the command to remove more than one Red Hat OpenShift Dedicated, for example, **FAILED_OSD_IDS=0,1,2**.

Remove the failed Red Hat OpenShift Dedicated:

```
oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> FORCE_OSD_REMOVAL=true | oc create -n openshift-storage -f -
```

Example output:

```
[root@fu71-f09-vm3 ~]# oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=0
FORCE_OSD_REMOVAL=true | oc create -n openshift-storage -f -
Warning: would violate PodSecurity "restricted:latest": allowPrivilegeEscalation != false (container "operator"
must set securityContext.allowPrivilegeEscalation=false), unrestricted capabilities (container "operator" must
set securityContext.capabilities.drop=["ALL"]), runAsNonRoot != true (pod or container "operator" must set
securityContext.runAsNonRoot=true), seccompProfile (pod or container "operator" must set
securityContext.seccompProfile.type to "RuntimeDefault" or "Localhost")
job.batch/ocs-osd-removal-job created
```

Check the status of the **ocs-osd-removal-job** pod to verify whether the Red Hat OpenShift Dedicated got removed successfully. A status of Completed confirms that the Red Hat OpenShift Dedicated removal job succeeded.

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

Example output:

```
[root@fu71-f09-vm3 ~]# oc get pod -n openshift-storage  | grep ocs-osd-removal
ocs-osd-removal-job-ls65l                                         0/1     Completed   0          23s
```

Ensure that the Red Hat OpenShift Dedicated removal is completed:

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

Example output:

```
[root@fu71-f09-vm3 ~]# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i
'completed removal'
2022-11-24 15:19:28.750910 I | cephosd: completed removal of OSD 0
```

Delete the **ocs-osd-removal-job**:

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

Delete the **Released PV** which is attached to the previous node.

```
oc get pv | grep -i released
```

```
local-pv-e4a12175                          2Ti        RWO            Delete           Released    openshift-
storage/ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m6gnz   ibm-spectrum-fusion-local              3h34m
[root@fu71-f09-vm3 ~]# oc delete pv local-pv-e4a12175
persistentvolume "local-pv-e4a12175" deleted
```

7. Recover the failed objects.
   a. Restart the mon deployment/pod:
      i. Update the nodeSelector in deployment with new node.

```
oc edit deployment -n openshift-storage rook-ceph-mon-d
```

```
nodeSelector:
        kubernetes.io/hostname: f09-prc4m-worker-cluster-b-djplp <<--new node
```

      ii. Scale the replica to 1 and wait till the mon pods are in running state.

```
oc scale deployment rook-ceph-mon-d --replicas=1 -n openshift-storage
```

Example:

```
oc scale deployment rook-ceph-mon-d --replicas=1 -n openshift-storage
    deployment.apps/rook-ceph-mon-d scaled
```

```
[root@fu71-f09-vm3 ~]# oc get pod -n openshift-storage | grep mon
    rook-ceph-mon-a-5bbb9dd98b-z54fx                                  2/2     Running     0          4m45s
    rook-ceph-mon-b-7fdd8f958b-lk9g2                                  2/2     Running     0          5m18s
    rook-ceph-mon-d-6945fbbfc5-nhhw8                                  2/2     Running     0          5m41s
```

   b. Verify the Red Hat OpenShift Dedicated pods.

Wait till all the pods are in running state.

```
oc get pods -o wide -n openshift-storage| grep osd
```

```
[root@fu71-f09-vm3 ~]# oc get pods -o wide -n openshift-storage| grep osd
rook-ceph-osd-0-d559cc4fb-xspr8                                   2/2     Running     0          4m58s
10.129.6.49       f09-prc4m-worker-cluster-b-djplp   <none>           <none>
rook-ceph-osd-1-6df7f9c669-n94md                                  2/2     Running     0          5m20s
10.128.6.95       f09-prc4m-worker-cluster-d-r5bxx   <none>           <none>
rook-ceph-osd-2-5c5d48ff7c-sdd7l                                  2/2     Running     0          5m17s
10.129.4.125      f09-prc4m-worker-cluster-c-mfb77   <none>           <none>
rook-ceph-osd-prepare-1bf7dd3d71fe899383e625dd0c27ea37-x9vtk      0/1     Completed   0          4h8m
10.128.6.79       f09-prc4m-worker-cluster-d-r5bxx   <none>           <none>
rook-ceph-osd-prepare-24272b5641dc95baffc7932d78894e3c-zhz8m      0/1     Completed   0          5m24s
10.129.6.48       f09-prc4m-worker-cluster-b-djplp   <none>           <none>
rook-ceph-osd-prepare-6f3c3b4626ec9888b6fbe5597afd55ea-zh7cp      0/1     Completed   0          4h8m
10.129.4.113      f09-prc4m-worker-cluster-c-mfb77   <none>           <none>
```

c. Verify the Red Hat OpenShift Dedicated encryption settings. If cluster wide encryption is enabled, make sure the “crypt” keyword beside the ocs-deviceset name(s)

```
oc debug node/<new-node-name> -- chroot /host dmsetup ls
```

Example output:

```
# oc debug node/fu47 -- chroot /host dmsetup ls
Starting pod/fu47-debug ...
To use host binaries, run `chroot /host`
ocs-deviceset-sc-lvs-0-data-0clwxf-block-dmcrypt        (253:0)
```

If verification fails, contact IBM support.

8. Exit maintenance mode after all steps are completed.

```
oc label odfclusters.odf.isf.ibm.com -n ibm-spectrum-fusion-ns odfcluster "odf.isf.ibm.com/maintenanceMode-"
```

Example output:

```
[root@fu40 ~]# oc label odfclusters.odf.isf.ibm.com -n ibm-spectrum-fusion-ns odfcluster
"odf.isf.ibm.com/maintenanceMode-"
odfcluster.odf.isf.ibm.com/odfcluster unlabeled
```

9. Go to Data foundation page in IBM Fusion user interface and check the health of the Storage cluster in the Health section.

Disk replacement and OpenShift dedicated OSD failure recovery

Learn how to replace a faulty disk in a Red Hat® OpenShift® dedicated Object Storage Device (OSD) cluster.

Procedure

1. Identify the failed OSD.
Command example:

```
oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

An example output where osd-19 is failed and the status is **CrashLoopBackOff**:

```
NAME                                READY   STATUS             RESTARTS        AGE     IP             NODE
rook-ceph-osd-0-85fcb5fd9-5cvws     2/2     Running            0               9d      9.42.107.150   compute-1-ru7.isf-racka.
rook-ceph-osd-1-5dd8bc8d9d-cbz7g    2/2     Running            0               37d     9.42.107.149   compute-1-ru6.isf-racka.
rook-ceph-osd-10-7cc49487b5-bdw6w   2/2     Running            0               9d      9.42.107.150   compute-1-ru7.isf-racka.
rook-ceph-osd-11-84cd6fb7d7-xpn22   2/2     Running            0               37d     9.42.107.148   compute-1-ru5.isf-racka.
rook-ceph-osd-12-7bb579498c-25tzh   2/2     Running            0               9d      9.42.107.150   compute-1-ru7.isf-racka.
rook-ceph-osd-13-866dbc7f57-kqqpn   2/2     Running            0               37d     9.42.107.149   compute-1-ru6.isf-racka.
rook-ceph-osd-14-6f7f6dd89b-7skg8   2/2     Running            0               37d     9.42.107.148   compute-1-ru5.isf-racka.
rook-ceph-osd-15-697c5b8577-9plff   2/2     Running            0               6h58m   9.42.107.146   control-1-ru3.isf-racka.
rook-ceph-osd-16-8d78df98c-khbh7    2/2     Running            0               37d     9.42.107.145   control-1-ru2.isf-racka.
rook-ceph-osd-17-c8ffbb5bf-q4xqb    2/2     Running            0               37d     9.42.107.145   control-1-ru2.isf-racka.
rook-ceph-osd-18-65f847c8d4-zpz2f   2/2     Running            0               37d     9.42.107.145   control-1-ru2.isf-racka.
rook-ceph-osd-19-85d475d68d-5kvmq   1/2     CrashLoopBackOff   6 (2m10s ago)   37d     9.42.107.145   control-1-ru2.isf-racka.
.
.
.
 <none>           <none>
rook-ceph-osd-7-7f79d4b76-jxz8x     2/2     Running            0               37d     9.42.107.149   compute-1-ru6.isf-racka.
rook-ceph-osd-8-75497d6999-wz296    2/2     Running            0               37d     9.42.107.148   compute-1-ru5.isf-racka.
rook-ceph-osd-9-56d5cc7c59-w29n2    2/2     Running            0               37d     9.42.107.149   compute-1-ru6.isf-racka.
```

2. Scale down **ocs-operator** first, then **rook-ceph-operator**.
   a. Scale down **ocs-operator**.
      i. Log in to the Red Hat OpenShift Container Platform web console and select your project.
      ii. Go to Workloads > Pods page and verify that **ocs-operator** is in the **Running** status in the Pods list.
      iii. Go to Workloads > Deployments page.
      iv. Click ocs-operator and check the Deployment details tab.

      For example, the Deployment details tab displays that **ocs-operator** has 1 Pod.

      v. Scale down the Pod number to 0.

vi. Go back to the Pods tab and check for the **ocs-operator**.

The **ocs-operator** does not appear or is no longer available in the Pods list.

b. Scale down **rook-ceph-operator**.

i. Go to Workloads > Pods page and verify that **rook-ceph-operator** is in the **Running** status in the Pods list.
ii. Go to Workloads > Deployments page.
iii. Click rook-ceph-operator and check the Deployment details tab.

For example, the Deployment details tab displays that **rook-ceph-operator** has 1 Pod.

iv. Scale down the Pod number to 0.
v. Go back to the Pods tab and check for the **rook-ceph-operator**.

The **rook-ceph-operator** does not appear or is no longer available in the Pods list.

3. Optional: Verify whether Pods for **ocs-operator** and **rook-ceph-operator** are removed as expected.
Command example:

```
oc get pods |grep operator
```

Example output:

```
noobaa-operator-79446cc789-gj66l                                  1/1     Running            0              38d
ocs-client-operator-console-7d85dc6bf9-vdbhn                      1/1     Running            0              9d
ocs-client-operator-controller-manager-85bbcc7bfd-znvx4           2/2     Running            1 (6d1h ago)   37d
odf-operator-controller-manager-59465654c-jmj5z                   2/2     Running            0              38d
```

4. Clean up failed OSD.

a. Scale down the **rook-ceph-osd** deployment.
Command example:

```
osd_id_to_remove=<replace-it-with-osd-id>
oc scale -n openshift-storage deployment rook-ceph-osd-${osd_id_to_remove} --replicas=0
```

Example output:

```
deployment.apps/rook-ceph-osd-19 scaled
```

b. Verify that the **rook-ceph-osd** Pod for the faulty OSD is deleted.
Command example:

```
oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

Example output:

```
NAME                                READY   STATUS    RESTARTS   AGE     IP             NODE
rook-ceph-osd-0-85fcb5fd9-5cvws     2/2     Running   0          9d      9.42.107.150   compute-1-ru7.isf-racka.rtp.ralei
rook-ceph-osd-1-5dd8bc8d9d-cbz7g    2/2     Running   0          37d     9.42.107.149   compute-1-ru6.isf-racka.rtp.ralei
rook-ceph-osd-10-7cc49487b5-bdw6w   2/2     Running   0          9d      9.42.107.150   compute-1-ru7.isf-racka.rtp.ralei
rook-ceph-osd-11-84cd6fb7d7-xpn22   2/2     Running   0          37d     9.42.107.148   compute-1-ru5.isf-racka.rtp.ralei
rook-ceph-osd-12-7bb579498c-25tzh   2/2     Running   0          9d      9.42.107.150   compute-1-ru7.isf-racka.rtp.ralei
rook-ceph-osd-13-866dbc7f57-kqqpn   2/2     Running   0          37d     9.42.107.149   compute-1-ru6.isf-racka.rtp.ralei
rook-ceph-osd-14-6f7f6dd89b-7skg8   2/2     Running   0          37d     9.42.107.148   compute-1-ru5.isf-racka.rtp.ralei
rook-ceph-osd-15-697c5b8577-9plff   2/2     Running   0          7h16m   9.42.107.146   control-1-ru3.isf-racka.rtp.ralei
rook-ceph-osd-16-8d78df98c-khbh7    2/2     Running   0          37d     9.42.107.145   control-1-ru2.isf-racka.rtp.ralei
rook-ceph-osd-17-c8ffbb5bf-q4xqb    2/2     Running   0          37d     9.42.107.145   control-1-ru2.isf-racka.rtp.ralei
rook-ceph-osd-18-65f847c8d4-zpz2f   2/2     Running   0          37d     9.42.107.145   control-1-ru2.isf-racka.rtp.ralei
rook-ceph-osd-2-5fd5548c56-gclrv    2/2     Running   0          37d     9.42.107.148   compute-1-ru5.isf-racka.rtp.ralei
rook-ceph-osd-20-b88868db5-h55h9    2/2     Running   0          37d     9.42.107.147   control-1-ru4.isf-racka.rtp.ralei
.
.
.
.
rook-ceph-osd-8-75497d6999-wz296    2/2     Running   0          37d     9.42.107.148   compute-1-ru5.isf-racka.rtp.ralei
rook-ceph-osd-9-56d5cc7c59-w29n2    2/2     Running   0          37d     9.42.107.149   compute-1-ru6.isf-racka.rtp.ralei
```

5. Remove the old OSD from the cluster.

a. Delete any existing **ocs-osd-removal** jobs, if present.
Command example:

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

b. Remove the old OSD from the cluster
Command example:

```
oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=${osd_id_to_remove} FORCE_OSD_REMOVAL=true |oc
create -n openshift-storage -f -
```

Important: Ensure that you set the correct **osd_id_to_remove**.
Example output:

```
job.batch/ocs-osd-removal-job created
```

c. Verify that the **ocs-osd-removal-job** has completed.
Command example:

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

Example output:

```
NAME                          READY   STATUS      RESTARTS   AGE
ocs-osd-removal-job-rh72h   0/1     Completed   0          39s
```

After the **ocs-osd-removal-job** completes, the associated Persistent Volumes (PVs) are released, and the Persistent Volume Claims (PVCs) are either deleted or move to the **Pending** state.

d. Check the PV state.
Command example:

```
oc get pv |grep Released
```

Example output:

```
local-pv-141605d2                          7153Gi     RWO            Delete           Released   openshift-storage/ocs-de
```

e. If encryption is enabled, remove the associated configuration.
Do the following steps:

i. Retrieve PVC and node details.

Command example:

```
oc describe pv local-pv-141605d2
```

Example output:

```
Name:              local-pv-141605d2
Labels:            kubernetes.io/hostname=control-1-ru2.isf-racka.rtp.raleigh.ibm.com
                   storage.openshift.com/owner-kind=LocalVolumeSet
                   storage.openshift.com/owner-name=ibm-spectrum-fusion-local
                   storage.openshift.com/owner-namespace=openshift-local-storage
Annotations:       pv.kubernetes.io/bound-by-controller: yes
                   pv.kubernetes.io/provisioned-by: local-volume-provisioner-control-1-ru2.isf-racka.rtp.raleigh.ibm
                   storage.openshift.com/device-id: nvme-MZWLJ7T6HALA-000V5_S5LLNE0R400096
                   storage.openshift.com/device-name: nvme1n1
Finalizers:        [kubernetes.io/pv-protection]
StorageClass:      ibm-spectrum-fusion-local
Status:            Released
Claim:             openshift-storage/ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465
Reclaim Policy:    Delete
Access Modes:      RWO
VolumeMode:        Block
Capacity:          7153Gi
Node Affinity:
  Required Terms:
    Term 0:        kubernetes.io/hostname in [control-1-ru2.isf-racka.rtp.raleigh.ibm.com]
Message:
Source:
    Type:  LocalVolume (a persistent volume backed by local storage on a node)
    Path:  /mnt/local-storage/ibm-spectrum-fusion-local/nvme-MZWLJ7T6HALA-000V5_S5LLNE0R400096
Events:
  Type     Reason              Age                  From     Message
  ----     ------              ----                 ----     -------
  Warning  VolumeFailedDelete  15s (x15 over 3m20s)  deleter  Error cleaning PV "local-pv-141605d2": failed to get v
```

ii. Remove the **dm-crypt**–managed device-mapper mapping from the OSD devices.

Command example:

```
oc debug node/<node name
chroot /host
dmsetup ls| grep <pvc name>
```

Example output:

```
ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt    (253:3)
```

iii. Remove the mapped device.

Command example:

```
cryptsetup luksClose --debug --verbose ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt
```

Example output:

```
# cryptsetup 2.6.0 processing "cryptsetup luksClose --debug --verbose ocs-deviceset-ibm-spectrum-fusion-local-0-data
# Verifying parameters for command close.
# Running command close.
# Installing SIGINT/SIGTERM handler.
# Unblocking interruption on signal.
# Allocating crypt device context by device ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt.
# Initialising device-mapper backend library.
# dm version   [ opencount flush ]   [16384] (*1)
# dm versions   [ opencount flush ]   [16384] (*1)
# Detected dm-ioctl version 4.48.0.
# Detected dm-crypt version 1.24.0.
# Device-mapper backend running with UDEV support enabled.
# dm status ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt  [ opencount noflush ]   [16384] (*
# Releasing device-mapper backend.
# Allocating context for crypt device (none).
# Initialising device-mapper backend library.
Underlying device for crypt device ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt disappeared.
# dm versions   [ opencount flush ]   [16384] (*1)
```

```
# dm table ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt  [ opencount flush securedata ]   [1
# dm versions   [ opencount flush ]   [16384] (*1)
# dm deps ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt  [ opencount flush ]   [16384] (*1)
# LUKS device header not available.
# Deactivating volume ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt.
# dm versions   [ opencount flush ]   [16384] (*1)
# dm status ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt  [ opencount noflush ]   [16384] (*
# dm versions   [ opencount flush ]   [16384] (*1)
# dm table ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt  [ opencount flush securedata ]   [1
# dm versions   [ opencount flush ]   [16384] (*1)
# dm deps ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt  [ opencount flush ]   [16384] (*1)
# dm versions   [ opencount flush ]   [16384] (*1)
# dm table ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt  [ opencount flush securedata ]   [1
# dm versions   [ opencount flush ]   [16384] (*1)
# Udev cookie 0xd4d2da5 (semid 0) created
# Udev cookie 0xd4d2da5 (semid 0) incremented to 1
# Udev cookie 0xd4d2da5 (semid 0) incremented to 2
# Udev cookie 0xd4d2da5 (semid 0) assigned to REMOVE task(2) with flags DISABLE_LIBRARY_FALLBACK         (0x20)
# dm remove ocs-deviceset-ibm-spectrum-fusion-local-0-data-32b6465-block-dmcrypt  [ opencount flush retryremove ]
# Udev cookie 0xd4d2da5 (semid 0) decremented to 0
# Udev cookie 0xd4d2da5 (semid 0) waiting for zero
# Udev cookie 0xd4d2da5 (semid 0) destroyed
# Releasing crypt device empty context.
# Releasing device-mapper backend.
Command successful.
```

f. Identify and delete the failed PV.
Do the following steps:

i. Identify the failed PV.

Command example:

```
oc get pv -l kubernetes.io/hostname=control-1-ru2.isf-racka.rtp.raleigh.ibm.com
```

The status of the failed PV shows as **Released**.

Example output:

```
NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS     CLAIM
local-pv-141605d2   7153Gi     RWO            Delete           Released   openshift-storage/ocs-deviceset-ibm-spectr
local-pv-4ce100b9   7153Gi     RWO            Delete           Bound      openshift-storage/ocs-deviceset-ibm-spectr
local-pv-5b320962   7153Gi     RWO            Delete           Bound      openshift-storage/ocs-deviceset-ibm-spectr
local-pv-6ba1664e   7153Gi     RWO            Delete           Bound      openshift-storage/ocs-deviceset-ibm-spectr
local-pv-89429641   7153Gi     RWO            Delete           Bound      openshift-storage/ocs-deviceset-ibm-spectr
local-pv-a3bb3040   7153Gi     RWO            Delete           Bound      openshift-storage/ocs-deviceset-ibm-spectr
local-pv-e9ed407b   7153Gi     RWO            Delete           Bound      openshift-storage/ocs-deviceset-ibm-spectr
```

ii. Delete the failed PV.

Command example:

```
oc delete pv local-pv-141605d2
```

Example output:

```
persistentvolume "local-pv-141605d2" deleted
```

iii. Verify that the failed PV is deleted.

Command example:

```
oc get pv -l kubernetes.io/hostname=control-1-ru2.isf-racka.rtp.raleigh.ibm.com
```

Example output:

```
NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
local-pv-4ce100b9   7153Gi     RWO            Delete           Bound    openshift-storage/ocs-deviceset-ibm-spectrum
local-pv-5b320962   7153Gi     RWO            Delete           Bound    openshift-storage/ocs-deviceset-ibm-spectrum
local-pv-6ba1664e   7153Gi     RWO            Delete           Bound    openshift-storage/ocs-deviceset-ibm-spectrum
local-pv-89429641   7153Gi     RWO            Delete           Bound    openshift-storage/ocs-deviceset-ibm-spectrum
local-pv-a3bb3040   7153Gi     RWO            Delete           Bound    openshift-storage/ocs-deviceset-ibm-spectrum
local-pv-e9ed407b   7153Gi     RWO            Delete           Bound    openshift-storage/ocs-deviceset-ibm-spectrum
```

6. Verify that the stale OSD entry is removed.
Command example:

```
oc get pods
```

Example output:

```
NAME                                                              READY   STATUS      RESTARTS       AGE
csi-addons-controller-manager-585444c85d-vlklj                    2/2     Running     0              38d
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-5rc9r            2/2     Running     0              37d
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-6zvlj            2/2     Running     0              37d
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-8rpcc            2/2     Running     0              37d
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-9xtqk            2/2     Running     4 (46h ago)    37d
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-k6jf9            2/2     Running     0              37d
openshift-storage.cephfs.csi.ceph.com-nodeplugin-796d9bb95        5/5     Running     0              37d
openshift-storage.cephfs.csi.ceph.com-nodeplugin-796d9bc86        5/5     Running     3 (9d ago)     9d
openshift-storage.cephfs.csi.ceph.com-ctrlplugin-srk79            2/2     Running     0              37d
openshift-storage.rbd.csi.ceph.com-ctrlplugin-dzfln               3/3     Running     0              37d
openshift-storage.rbd.csi.ceph.com-ctrlplugin-j5jk5               3/3     Running     5 (46h ago)    37d
openshift-storage.rbd.csi.ceph.com-ctrlplugin-lw5lt               3/3     Running     0              37d
openshift-storage.rbd.csi.ceph.com-ctrlplugin-mf6pp               3/3     Running     0              37d
```

```
openshift-storage.rbd.csi.ceph.com-ctrlplugin-nc9g9               3/3     Running     0              37d
openshift-storage.rbd.csi.ceph.com-nodeplugin-7cf4b47dd9-89j27    6/6     Running     0              37d
openshift-storage.rbd.csi.ceph.com-nodeplugin-7cf4b47dd9-9w4      6/6     Running     0              37d
openshift-storage.rbd.csi.ceph.com-ctrlplugin-tb2wf               3/3     Running     0              37d
noobaa-core-0                                                     2/2     Running     0              37d
noobaa-db-pg-0                                                    1/1     Running     0              37d
noobaa-endpoint-68c99f7ddf-vlfbs                                  1/1     Running     0              37d
noobaa-operator-79446cc789-gj66l                                  1/1     Running     0              38d
ocs-client-operator-console-7d85dc6bf9-vdbhn                      1/1     Running     0              9d
ocs-client-operator-controller-manager-85bbcc7bfd-znvx4           2/2     Running     1 (6d1h ago)   37d
ocs-metrics-exporter-858847c98f-m6x4l                             1/1     Running     0              37d
ocs-osd-removal-job-rh72h                                         0/1     Completed   0              22m
ocs-provider-server-6d49cffc97-q4vfm                              1/1     Running     0              37d
odf-console-69bdbf8bc4-xt7jh                                      1/1     Running     0              38d
odf-operator-controller-manager-59465654c-jmj5z                   2/2     Running     0              38d
rook-ceph-crashcollector-18f4004ce1313cc06e57f6eb215a1321-lhpn6   1/1     Running     0              37d
rook-ceph-crashcollector-3f4d91760a68a6dbc7579b3e8b5915dd-m84jq   1/1     Running     0              8d
rook-ceph-crashcollector-6ff2efb9f9fe11fb194447b2fbf5c1e4-85ztt   1/1     Running     0              9d
rook-ceph-crashcollector-74af1983e44d54775407e88bf7382e9e-vchft   1/1     Running     0              37d
rook-ceph-crashcollector-9178b6908700ea615c9dc0b5706acedea-84kcs  1/1     Running     0              37d
rook-ceph-crashcollector-e67f06af5747db0daafd2ddc1be00528-wlkn8   1/1     Running     0              46h
rook-ceph-exporter-compute-1-ru5.isf-racka.rtp.raleigh.ibm7p7df   1/1     Running     0              37d
rook-ceph-exporter-compute-1-ru6.isf-racka.rtp.raleigh.ibmz7qk8   1/1     Running     0              37d
rook-ceph-exporter-compute-1-ru7.isf-racka.rtp.raleigh.ibm4v9ct   1/1     Running     0              8d
rook-ceph-exporter-control-1-ru2.isf-racka.rtp.raleigh.ibmrd6f9   1/1     Running     0              9d
rook-ceph-exporter-control-1-ru3.isf-racka.rtp.raleigh.ibms5wxb   1/1     Running     0              46h
rook-ceph-exporter-control-1-ru4.isf-racka.rtp.raleigh.ibmtq9dr   1/1     Running     0              37d
rook-ceph-mds-ocs-storagecluster-cephfilesystem-a-6b557764fgdv7   2/2     Running     10 (46h ago)   37d
rook-ceph-mds-ocs-storagecluster-cephfilesystem-b-6d8c97d72fjts   2/2     Running     10 (46h ago)   9d
rook-ceph-mgr-a-864fb44895-pl2n8                                  3/3     Running     0              46h
rook-ceph-mgr-b-7cf6b49bcd-q5k49                                  3/3     Running     0              37d
rook-ceph-mon-a-79897dd546-hbstb                                  2/2     Running     0              37d
rook-ceph-mon-c-bf487b7c7-flcpf                                   2/2     Running     0              37d
rook-ceph-mon-d-76f96d4fd4-h4h9t                                  2/2     Running     0              46h
rook-ceph-osd-0-85fcb5fd9-5cvws                                   2/2     Running     0              9d
rook-ceph-osd-1-5dd8bc8d9d-cbz7g                                  2/2     Running     0              37d
rook-ceph-osd-10-7cc49487b5-bdw6w                                 2/2     Running     0              9d
rook-ceph-osd-11-84cd6fb7d7-xpn22                                 2/2     Running     0              37d
rook-ceph-osd-12-7bb579498c-25tzh                                 2/2     Running     0              9d
rook-ceph-osd-13-866dbc7f57-kqqpn                                 2/2     Running     0              37d
rook-ceph-osd-14-6f7f6dd89b-7skg8                                 2/2     Running     0              37d
rook-ceph-osd-15-697c5b8577-9plff                                 2/2     Running     0              7h41m
rook-ceph-osd-16-8d78df98c-khbh7                                  2/2     Running     0              37d
rook-ceph-osd-17-c8ffbb5bf-q4xqb                                  2/2     Running     0              37d
rook-ceph-osd-18-65f847c8d4-zpz2f                                 2/2     Running     0              37d
rook-ceph-osd-2-5fd5548c56-gclrv                                  2/2     Running     0              37d
rook-ceph-osd-20-b88868db5-h55h9                                  2/2     Running     0              37d
rook-ceph-osd-21-556947cb6f-k29mm                                 2/2     Running     0              37d
rook-ceph-osd-22-5d79578d54-b9qlb                                 2/2     Running     0              37d
rook-ceph-osd-23-84bcb674c5-6jzxx                                 2/2     Running     0              37d
rook-ceph-osd-24-6df4949c8c-qvczd                                 2/2     Running     0              37d
rook-ceph-osd-26-5d7b7b6f46-vqgz4                                 2/2     Running     0              46h
rook-ceph-osd-27-59b8c85b78-wz4pb                                 2/2     Running     0              46h
rook-ceph-osd-28-54dc8db898-ml8r5                                 2/2     Running     0              46h
rook-ceph-osd-29-5666b4975c-qrpm5                                 2/2     Running     0              46h
rook-ceph-osd-3-7cfc7899dc-px8xm                                  2/2     Running     0              9d
rook-ceph-osd-30-7cc88bfcfc-xsnvc                                 2/2     Running     0              37d
rook-ceph-osd-31-78cc567d99-8jhkh                                 2/2     Running     0              37d
rook-ceph-osd-32-6c5fcdfb45-4jz2q                                 2/2     Running     0              46h
rook-ceph-osd-33-694f48d4f8-4sflf                                 2/2     Running     0              22d
rook-ceph-osd-4-6f4fb99d7-w7vrb                                   2/2     Running     0              37d
rook-ceph-osd-5-7b7ff5cfb5-blpj7                                  2/2     Running     0              37d
rook-ceph-osd-6-5d9856b54f-bjpmm                                  2/2     Running     0              9d
rook-ceph-osd-7-7f79d4b76-jxz8x                                   2/2     Running     0              37d
rook-ceph-osd-8-75497d6999-wz296                                  2/2     Running     0              37d
rook-ceph-osd-9-56d5cc7c59-w29n2                                  2/2     Running     0              37d
rook-ceph-rgw-ocs-storagecluster-cephobjectstore-a-668df44tbvtn   2/2     Running     0              37d
rook-ceph-tools-68d744d548-jwj5l                                  1/1     Running     0              23d
storageclient-737342087af10580-status-reporter-29061474-tv52t     0/1     Completed   0              51s
ux-backend-server-7796f96896-kwtsf                                2/2     Running     0              38d
```

7. Add a new disk to the node.
   Important:
   - If a fresh or unused disk is inserted, a new PV is created in the **`Available`** state.
   - If a used disk is inserted as a replacement, it must be formatted for the corresponding PV to become available.

   After adding the new disk, verify the status of the new PV.

   Command example:

   ```
   oc get pv |grep local
   ```

   Example output:

```
local-pv-159e1b8e                          7153Gi     RWO            Delete           Bound       openshift-storage/ocs-device
local-pv-2ac8a85b                          7153Gi     RWO            Delete           Bound       openshift-storage/ocs-device
local-pv-406f1af8                          7153Gi     RWO            Delete           Bound       openshift-storage/ocs-device
local-pv-451f6341                          7153Gi     RWO            Delete           Bound       openshift-storage/ocs-device
local-pv-4ce100b9                          7153Gi     RWO            Delete           Bound       openshift-storage/ocs-device
local-pv-4da6a449                          7153Gi     RWO            Delete           Available
local-pv-521ee04f                          7153Gi     RWO            Delete           Bound       openshift-storage/ocs-device
.
.
.
.
local-pv-b7c4932c                          7153Gi     RWO            Delete           Bound       openshift-storage/ocs-device
```

8. Scale up **ocs-operator** first, then **rook-ceph-operator**.
    a. Scale up **ocs-operator**.
        i. Go to Workloads > Deployments page.
        ii. Click ocs-operator and check the Deployment details tab.

            For example, the Deployment details tab displays that **ocs-operator** has 0 Pod.

        iii. Scale up the Pod number to 1.
        iv. Go to Workloads > Pods page and verify that **ocs-operator** is in the **Running** status in the Pods list.
    b. Scale up **rook-ceph-operator**.
        i. Go to Workloads > Deployments page.
        ii. Click rook-ceph-operator and check the Deployment details tab.

            For example, the Deployment details tab displays that **rook-ceph-operator** has 0 Pod.

        iii. Scale up the Pod number to 1.
        iv. Go to Workloads > Pods page and verify that **rook-ceph-operator** is in the **Running** status in the Pods list.
9. Verify that the new OSD Pod is running.
   Command example:

   ```
   oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
   ```

   Example output:

   ```
   NAME                                READY   STATUS    RESTARTS   AGE    IP             NODE
   rook-ceph-osd-0-85fcb5fd9-5cvws     2/2     Running   0          9d     9.42.107.150   compute-1-ru7.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-1-5dd8bc8d9d-cbz7g    2/2     Running   0          37d    9.42.107.149   compute-1-ru6.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-10-7cc49487b5-bdw6w   2/2     Running   0          9d     9.42.107.150   compute-1-ru7.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-11-84cd6fb7d7-xpn22   2/2     Running   0          37d    9.42.107.148   compute-1-ru5.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-12-7bb579498c-25tzh   2/2     Running   0          9d     9.42.107.150   compute-1-ru7.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-13-866dbc7f57-kqqpn   2/2     Running   0          37d    9.42.107.149   compute-1-ru6.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-14-6f7f6dd89b-7skg8   2/2     Running   0          37d    9.42.107.148   compute-1-ru5.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-15-697c5b8577-9plff   2/2     Running   0          8h     9.42.107.146   control-1-ru3.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-16-8d78df98c-khbh7    2/2     Running   0          37d    9.42.107.145   control-1-ru2.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-17-c8ffbb5bf-q4xqb    2/2     Running   0          37d    9.42.107.145   control-1-ru2.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-18-65f847c8d4-zpz2f   2/2     Running   0          37d    9.42.107.145   control-1-ru2.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-19-74cc9c8bc6-cc8p4   2/2     Running   0          111s   9.42.107.145   control-1-ru2.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-2-5fd5548c56-gclrv    2/2     Running   0          37d    9.42.107.148   compute-1-ru5.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-20-b88868db5-h55h9    2/2     Running   0          37d    9.42.107.147   control-1-ru4.isf-racka.rtp.raleigh.ibm
   .
   .
   .
   rook-ceph-osd-8-75497d6999-wz296    2/2     Running   0          37d    9.42.107.148   compute-1-ru5.isf-racka.rtp.raleigh.ibm
   rook-ceph-osd-9-56d5cc7c59-w29n2    2/2     Running   0          37d    9.42.107.149   compute-1-ru6.isf-racka.rtp.raleigh.ibm
   ```

10. Verify that the PV and PVC are successfully created.
    Command example for PVC:

    ```
    oc get pvc -n openshift-storage |grep local-pv-4da6a449
    ```

    Command example for PV:

    ```
    oc get pv -n openshift-storage |grep local-pv-4da6a449
    ```

    Example output for PVC:

    ```
    ocs-deviceset-ibm-spectrum-fusion-local-0-data-35tw7jf   Bound    local-pv-4da6a449                          7153Gi     RWO
    ```

    Example output for PV:

    ```
    local-pv-4da6a449                          7153Gi     RWO            Delete           Bound    openshift-storage/ocs-deviceset
    ```

Results

Faulty disk or failed OSD replacement is complete.

What to do next

Wait for the storage cluster to reconcile its health status. If it remains in the **Warning** state despite all Fusion Data Foundation Pods running, check for and resolve any previous crashes.

To verify the health status, do as follows:

- Log in to the IBM Fusion web console and go to Storage > Local storage to check the health of the local storage.
- In the Red Hat OpenShift Container Platform web console, go to Storage > Data Foundation and check the health of the storage.

Storage client status reporter pod is stuck in terminating state

Learn how to diagnose and fix issues causing the storage client status reporter pod to get stuck during termination.

If the storageclient-status-reporter pod gets stuck during termination, the following options don't work:

- Force deletion of the pod.
- Patching of the finalizer.

To fix this issue, remove the finalizer by using the following command:

```
for pod in $(oc get pod --field-selector=status.phase=Failed -oname -nopenshift-storage | grep storageclient)
do
    oc patch $pod -nopenshift-storage --type merge -p '{"status":{"qosClass":"BestEffort"}}' --subresource status
    oc patch $pod -nopenshift-storage --type merge -p '{"metadata":{"finalizers": []}}'
done
```

Global Data Platform

Learn how to install and configure Global Data Platform service.

The deployment method for the Global Data Platform service depends on the version of Red Hat® OpenShift® Container Platform in your environment. Starting with Red Hat OpenShift Container Platform 4.21 and Fusion Data Foundation 4.21 or later, the service is integrated into Fusion Data Foundation. For more information, see Fusion Data Foundation service with Global Data Platform.
Important: This chapter applies to the configuration of the Global Data Platform service on Red Hat OpenShift Container Platform 4.20 or earlier.

- **Installing Global Data Platform**
  From the Services page, enable the IBM Fusion Global data platform service.
- **Global Data Platform service install and upgrade issues**
  Use this troubleshooting information to resolve install and upgrade problems that are related to Global Data Platform service.
- **Uninstalling Global Data Platform**
  Steps to uninstall Global Data Platform by using command line interface.
- **Configuring storage**
  You can configure Global Data Platform as your storage.
- **Setting up encryption for storage**
- **Troubleshooting Global Data Platform issues**
  Global Data Platform (GDP) storage issues on Red Hat OpenShift Container Platform 4.20 or earlier can include filesystem mount delays, CSI driver errors, and backup failures with static persistent volumes.

Installing Global Data Platform

From the Services page, enable the IBM Fusion Global data platform service.

Before you begin

Important:

- IBM Fusion Access for Storage Area Network (SAN), Global Data Platform, and Content-Aware Storage are mutually exclusive services. If IBM Fusion Access for SAN is planned or installed, you cannot deploy the Global Data Platform or Content-Aware Storage services. Review your architectural requirements and deployment goals before installing these services.
- Check your Red Hat® OpenShift® Container Platform version:
  - **Version 4.20 or earlier**: Continue with this procedure
  - **Version 4.21 or later**: see Fusion Data Foundation service with Global Data Platform

About this task

Ensure that you are aware of the following points before you install Global Data Platform:

- For more information about deployment platforms and their supported services in IBM Fusion, see Support matrix.
- The IBM Fusion supports Global Data Platform version 6.0.0.2 or earlier.
- Do NOT delete IBM Storage Scale pods. In many circumstances, the deletion of Scale pods has implications on the availability and data integrity.
- This service enablement applies Machine Config Operator changes in Red Hat OpenShift, and as a result, all compute nodes get restarted.
- To enable GDP on the infra node, manually create the `ScaleCluster` CR and set the infraNode:true. For more information, see Configuring Global Data Platform service on infrastructure nodes.
- If Global Data Platform service is installed on a OpenShift Container Platform cluster, check whether the nodes are running with secure boot:

  ```
  mokutil --sb-state
  ```

  As a root user, disable secure boot with the following command:

  ```
  mokutil --disable-validation
  ```

Procedure

1. Go to the Services page in the IBM Fusion user interface.
2. In the Available section, click Global data platform tile.
3. In the Global Data Platform page, go through the details and click Install.
   After you enable the Global Data Platform, you can view the service version and health status.

Table 1. Health states Global Data Platform service

State	Description

State	Description
`Healthy`	The `Running` and `Ready` states for the following items indicate that the service is healthy: • All the pods in `ibm-spectrum-scale` namespace are running. • All the pods in `ibm-spectrum-scale-csi` namespace are running. • GPFS Scale service is active on all the pods. • The filesystem is mounted on all the nodes.
`Degraded`	If any of the following conditions are not met, then the service goes to a degraded state. • Any one of the scale core pod statuses is not running per Recovery Group (RG) • GPFS service is not active on any one of the pods per RG. • If one of the PM collector pods is not running or all containers are not in the ready state. • If one of the GUI pods is not running or all containers are not in the ready state.

From the ellipsis menu, you can download logs and view documentation. After you successfully collect the logs, a success notification gets displayed. The notification disappears automatically after some time.
If failures occur, go through the downloaded logs to understand the cause of the failure and fix the issue. For more information about service issues in IBM Fusion, see Global Data Platform service install and upgrade issues.

4. Verify Global Data Platform installation.

   Validate Global Data Platform operators, deployments, and pods from the OpenShift management console
   a. Log in to the OpenShift web console.
   b. From the menu navigation, click Operators > Installed Operators.
   c. From the Installed Operators list, click IBM Storage Fusion operator. The Details tab is open by default.
   d. From the menu navigation, click Workloads > Pods.
   e. Check whether the following namespace and its deployment is available:

      `ibm-spectrum-scale-operator` project
      `ibm-spectrum-scale-controller-manager`

   f. From the menu navigation, go to Workloads > Pods.
   g. Check whether you can see the following pods:

      `ibm-spectrum-scale-operator`
      `ibm-spectrum-scale-controller-manager`

5. Verify Global Data Platform configuration.

   Validate Global Data Platform pods from the OpenShift management console
   a. Log in to the OpenShift web console.
   b. From the menu navigation, go to Workloads > Pods.
   c. Check whether you can see the following pods:

      `ibm-spectrum-scale-csi` project
      `ibm-spectrum-scale-csi-provisioner`
      `ibm-spectrum-scale-csi-attacher`
      `ibm-spectrum-scale-csi`
      Note: For Global Data Platform version 6.0.0.1, the `ibm-spectrum-scale-csi-operator` pod is no longer available.
      `ibm-spectrum-scale` project
      `scale-core-pods`
      `ibm-spectrum-scale-gui-0`
      `ibm-spectrum-scale-gui-1`
      `ibm-spectrum-scale-pmcollector-0`
      `ibm-spectrum-scale-pmcollector-1`

   Validate other pods from the CLI
   a. To list all DNS pods, run the OC command:

      ```
      ./oc get pods -n ibm-spectrum-scale-dns
      ```

      Note: You can see pods under `ibm-spectrum-scale-dns` only after connecting to `FileSystem`.
      For example, see the following result:

      ```
      oc get po -n ibm-spectrum-scale-dns
      NAME             READY   STATUS    RESTARTS   AGE
      coredns-4gmrg    1/1     Running   3          7d16h
      coredns-bjpv7    1/1     Running   5          7d16h
      coredns-hnrph    1/1     Running   3          7d16h
      coredns-j4dxq    1/1     Running   3          7d16h
      coredns-ln6gj    1/1     Running   3          7d16h
      coredns-rtfdg    1/1     Running   3          7d16h
      ```

      Ensure all the pods are up and running.

   Verify the storage class.
   Note: Ensure that the following storage classes are created.

   ```
   oc get sc
   ```

What to do next

1. Go to Remote file systems page of the user interface and connect to remote file systems. For the procedure to connect to remote file systems, see Connecting to remote IBM Storage Scale file systems.
2. To consume the remote file system by using Global Data Platform, see Static provisioning of Persistent Volume in filesystem and Dynamic provisioning of Persistent Volume in filesystem.
3. You can now install other available IBM Fusion services.

Global Data Platform service install and upgrade issues

Use this troubleshooting information to resolve install and upgrade problems that are related to Global Data Platform service.

Warning: Do NOT delete IBM Storage Scale pods. Deletion of Scale pods in many circumstances has implications on availability and data integrity.
Important: This section applies to troubleshooting installation and upgrade issues related to the Global Data Platform service on Red Hat® OpenShift® Container Platform version 4.20 or earlier.

Cannot create or update the ScaleManager CR after an IBM Fusion upgrade

Symptoms

The Global Data Platform service does not have the Upgrade button on the IBM Fusion user interface after the IBM Fusion upgrade. It is because the status of ScaleManager CR cannot be updated. You might see the following error in the **isf-cns-operator** pod logs.
Example:

```
2024-06-26T10:48:36.304945925Z 2024-06-26T10:48:36.304Z ERROR   Reconciler error        {"controller": "scalemanager", "contro
```

The ScaleManager CR cannot be created after a IBM Fusion upgrade as the Global Data Platform installation gets stuck with the following error in the **isf-prereq-operator** pod logs:

```
2024-08-01T01:44:24Z    ERROR   controllers.prereq.SpectrumFusion       Failed to create ScaleManager CR:       {"fusionsdsclu
```

Cause

Somehow the webhooks for IBM Fusion HCI were applied to IBM Fusion software.

Resolution

1. In the OpenShift Container Platform console, go to Home > Search.
2. From the Resources list, select **MutatingWebhookConfiguration**.
3. Select the Label drop-down list and change it to `Name`.
4. Search for **isf-cns-operator**. Check whether there exists an instance of **isf-cns-operator-mutating-webhook-configuration** webhook. If it exists, take a backup of the older one and delete it.
5. Go back to Home > Search page.
6. From the Resources, select **ValidatingWebhookConfiguration**.
7. Search for **isf-cns-operator** and check whether there exists an instance of **isf-cns-operator-validating-webhook-configuration** webhook. If it exists, take the backup of the older one and delete it.

Upgrade does not complete as node pod is in `ContainerStatusUnknown` state

Problem statement

Global data platform upgrade does not complete because a compute node pod is in **ContainerStatusUnknown** state.

Workaround

Delete the pod in **ContainerStatusUnknown** state and restart the compute node.

Pod drain issues in Global data platform upgrade

Diagnosis and resolution

If the upgrade of Global data platform gets struck, then do the following diagnose and resolve:

1. Go through the logs from the scale operator.
2. Check whether you observe the following error:
   `ERROR Drain error when evicting pods`
3. If it is a drain error, then check whether the virtual machine's PVC were created ReadWriteOnce.
   The PVCs created by using ReadWriteAfter are not shareable or moveable and can cause the draining of the node.

   For example:

   oc get pods

NAME	READY	STATUS	RESTARTS	AGE
compute-0	2/2	Running	0	20h
compute-1	2/2	Running	0	24h
compute-13	2/2	Running	0	21h
compute-14	2/2	Running	3	21h
compute-2	2/2	Running	0	21h
compute-3	2/2	Running	0	21h
control-0	2/2	Running	0	23h
control-1	2/2	Running	0	23h
control-1-1	2/2	Running	0	23h
control1-1-reboot	0/1	ContainerStatusUnknown	1	23h
control-2	2/2	Running	0	23h
ibm-spectrum-scale-gui-0	4/4	Running	0	20h
ibm-spectrum-scale-gui-1	4/4	Running	0	23h
ibm-spectrum-scale-pmcollector-0	2/2	Running	0	23h
ibm-spectrum-scale-pmcollector-1	2/2	Running	0	20h

4. If virtual machine's PVC got created ReadWriteOnce, then stop the virtual machine to continue the upgrade of scale pods.
5. If the upgrade fails further, then check whether the reason can be due to an orphaned **control-1-reboot** container left in the **ibm-spectrum-scale** namespace. Delete the container to resume and complete the installation.

IBM Storage Scale might get stuck during upgrade

Resolution

Do the following steps:

1. Shut down all applications that use storage.
2. Scale down the IBM Storage Scale operator. Make the replica as 0 for deployment **ibm-spectrum-scale-controller-manager**.
   a. Use **ibm-spectrum-scale-operator** project:

   ```
   oc project ibm-spectrum-scale-operator
   ```

   b. Scale down the IBM Storage Scale operator.

   ```
   oc scale --replicas=0 deployment ibm-spectrum-scale-controller-manager -n ibm-spectrum-scale-operator
   ```

   c. Run the following command to confirm resources are not found in **ibm-spectrum-scale-operator** namespace.

   ```
   oc get po
   ```

3. Log in to the one of the IBM Storage Scale core pods and shut down the server.
   a. Log in to the server:

   ```
   oc rsh compute-0
   ```

   b. Shut down the server:

   ```
   mmshutdown -a
   ```

   Example output:

   ```
   Thu May 19 07:46:47 UTC 2022: mmshutdown: Starting force unmount of GPFS file systems
   Thu May 19 07:46:52 UTC 2022: mmshutdown: Shutting down GPFS daemons
   compute-13.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Shutting down!
   compute-14.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Shutting down!
   compute-0.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Shutting down!
   control-1.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Shutting down!
   control-0.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Shutting down!
   compute-1.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Shutting down!
   compute-2.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Shutting down!
   control-2.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Shutting down!
   compute-13.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  'shutdown' command about to kill process 1
   compute-13.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unloading modules from /lib/modules/4.18.0
   compute-14.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  'shutdown' command about to kill process 1
   compute-14.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unloading modules from /lib/modules/4.18.0
   compute-14.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  mmfsenv: Module mmfslinux is still in use.
   compute-13.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  mmfsenv: Module mmfslinux is still in use.
   compute-14.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unloading module mmfs26
   compute-13.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unloading module mmfs26
   compute-14.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unmount all GPFS file syste
   ..
   compute-2.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unloading module mmfs26
   control-2.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unloading module mmfs26
   control-2.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unloading module mmfslinux
   compute-0.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unloading module mmfslinux
   compute-2.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local:  Unloading module mmfslinux
   Thu May 19 07:47:03 UTC 2022: mmshutdown: Finished
   ```

4. Check whether the states are down:

   ```
   mmgetstate -a
   ```

   Sample output:

   ```
   Node number  Node name          GPFS state
   ------------------------------------------
              1  control-0-daemon   down
              2  control-1-daemon   down
              3  control-2-daemon   down
              4  compute-14-daemon  down
              5  compute-13-daemon  down
              6  compute-0-daemon   down
              7  compute-1-daemon   down
              8  compute-2-daemon   down
   ```

5. Delete all the pods in **ibm-spectrum-scale** namespace:

   ```
   oc delete pods --all -n ibm-spectrum-scale
   ```

6. To verify, get the list of pods:

   ```
   oc get po -n ibm-spectrum-scale
   ```

   Sample output:

   ```
   NAME                              READY   STATUS    RESTARTS   AGE
    ibm-spectrum-scale-gui-0          0/4     Pending   0          7s
    ibm-spectrum-scale-pmcollector-0  0/2     Pending   0          7s
   ```

   Note: Only IBM Storage Scale pods get deleted and not GUI and **pmcollector**.

CSI Pods experience scheduling problems post node drain

Problem statement
: Whenever a compute node is drained, an eviction of CSI PODs or sidecar PODs occur. The remaining available set of compute nodes cannot host the CSI PODs or sidecar PODs because of resource constraints at that point in time.

Resolution
: Ensure that a functional system exists with available compute nodes, having sufficient resources to accommodate evicted CSI PODs or sidecar PODs.

Global Data Platform upgrade progress

Problem statement
: During Global Data Platform upgrade, the progress percentage can go beyond 100% in a few cases. You can ignore this issue, as the percentage comes down to 100 % after the successful completion of the upgrade.

Known issues in upgrade

- During upgrade, the IBM Fusion rack user interface, IBM Storage Scale user interface, Grafana endpoint, and Applications are not reachable for sometime.
- During the upgrade, an intermittent error `-1 node updated successfully out of 5 nodes` shows on the IBM Fusion user interface in the upgrade details page.
- During the upgrade, an intermittent error shows on the IBM Fusion user interface that the progress percentage for the Global Data Platform decreased. It occurs, especially when the upgrade for one node is completed.
- During the upgrade, an intermittent error `Global Data Platform upgrade failed` shows on the IBM Fusion user interface in the upgrade details page. Ignore the error because it might recover during the next reconciliation.

Uninstalling Global Data Platform

Steps to uninstall Global Data Platform by using command line interface.

Before you begin

- Important:
  Check your Red Hat® OpenShift® Container Platform version:
  - **Version 4.20 or earlier**: Continue with this procedure
  - **Version 4.21 or later**: see Uninstalling Global Data Platform from Fusion Data Foundation external mode
- Run the following commands to define namespace environmental variables:

  ```
  export FUSION_NS=<Fusion namespace>
  export SCALE_NS=ibm-spectrum-scale
  ```

  Important: Replace `<Fusion namespace>` with your namespace.
- Ensure that the applications accessing the storage filesystem through IBM Storage Scale Container Storage Interface Driver, must be stopped before continuing.
- Do the following steps to remove all the applications:
  1. Stop all the applications that are accessing storage through the IBM Storage Scale Container Storage Interface Driver.
  2. Delete all the Persistent Volume Claims (PVCs) and Persistent Volumes (PVs) provisioned by IBM Storage Scale Container Storage Interface Driver.

Procedure

1. Do the following steps to delete the IBM Fusion storage CRs.
   a. Run the following commands to delete encryption settings, if you have:

   ```
   oc -n ${FUSION_NS} get EncryptionClient | grep -v NAME | awk '{print $1}' | xargs -I{} oc -n ${FUSION_NS} delete EncryptionClient {}
   oc -n ${FUSION_NS} get EncryptionServer | grep -v NAME | awk '{print $1}' | xargs -I{} oc -n ${FUSION_NS} delete EncryptionServer {}
   ```

   b. Run the following command to delete all the filesystems:

   ```
   oc -n ${FUSION_NS} get ScaleCluster | grep -v NAME | awk '{print $1}' | xargs -I{} oc -n ${FUSION_NS} delete ScaleCluster {}
   ```

   c. Run the following commands to verify all `EncryptionServer` and `EncryptionClient` CRs are deleted from IBM Fusion namespace:

   ```
   oc -n ${FUSION_NS} get EncryptionServer
   oc -n ${FUSION_NS} get EncryptionClient
   ```

   d. Run the following command to verify all the `ScaleCluster` CRs are deleted from IBM Fusion namespace.

   ```
   oc -n ${FUSION_NS} get ScaleCluster
   ```

   e. Run the following commands to delete the sample PV templates:

   ```
   oc get pv -o name | grep 'ibm-spectrum-scale-fs-.*-template' | xargs -I{} oc delete {}
   ```

   f. Run the following command to delete the `VolumeSnapshot` class:

   ```
   oc get volumesnapshotclass | grep spectrumscale.csi.ibm.com | awk '{print $1}' | xargs -I{} oc delete volumesnapshotclass {}
   ```

2. Do the following steps to delete IBM Storage Scale container native resources.

a. Run the following command and record the scale cluster name:

```
oc -n ${SCALE_NS} get daemon -o yaml | grep clusterName: | awk -F " " '{print $2}'
```

Sample scale cluster name:

```
ibm-storage-fusion-xx.xxx.example.com
```

b. Run the following command to delete the IBM Storage Scale container native cluster:

```
oc delete cluster.scale.spectrum.ibm.com ibm-spectrum-scale
```

c. Run the following commands to delete the sample Storage class, PVC, and PV:

```
oc -n ${SCALE_NS} get pvc -o name | grep pmcollector | xargs -I{} oc -n ${SCALE_NS} delete {}
oc delete sc -lapp.kubernetes.io/instance=ibm-spectrum-scale,app.kubernetes.io/name=pmcollector
oc delete pv -lapp.kubernetes.io/instance=ibm-spectrum-scale,app.kubernetes.io/name=pmcollector
```

d. Do the following steps to delete IBM Storage Scale container native directories on each worker nodes:

- Run the following command to list the nodes that have the `node-role.kubernetes.io/worker=` label.

```
oc get nodes -l 'node-role.kubernetes.io/worker=' -o jsonpath="{range .items[*]}{.metadata.name}{'\n'}"
```

- For each of the worker nodes listed, enter the following command to create a debug pod that removes the kernel modules and the host path volume mounted directories used by IBM Storage Scale container native:

```
oc debug node/<openshift_node> -T -- chroot /host sh -c "rm -rf /var/mmfs; rm -rf /var/adm/ras; rmmod tracedev
mmfs26 mmfslinux;"
```

- Ensure that none of the artifacts are left by entering the following validation command:

```
oc debug node/<openshift_node> -T -- chroot /host sh -c "ls /var/mmfs; ls /var/adm/ras; rmmod tracedev mmfs26
mmfslinux;"
```

e. Run the following commands to delete the node labels created by IBM Storage Scale container native operator:

```
oc label node --all scale.spectrum.ibm.com/role-
oc label node --all scale.spectrum.ibm.com/designation-
oc label node --all scale-
```

3. Do the following steps on the remote IBM Storage Scale cluster, to delete access permission:
Note: If your storage cluster is on AWS ROSA, skip this section and revoke the filesystem access from the storage cluster. For the procedure to revoke, see Revoke filesystem access from the storage cluster.

- Run the following command to query the name of the containerized client cluster:

```
mmauth show all | grep <scale_cluster_name>
```

Replace the `<scale_cluster_name>` with the actual storage cluster name obtained in step 2.a.

- Run the following command to remove the client cluster authorization:

```
mmauth delete ibm-storage-fusion-xx.xxx.example.com
```

4. Do the following steps to disable Global Data Platform.

a. Run the following command to disable IBM Fusion Global Data Platform service:
Command example:

```
oc -n ${FUSION_NS} patch spectrumfusion/spectrumfusion --type=merge --patch='{"spec":{"GlobalDataPlatform":
{"Enable":false}}}'
```

b. Run the following command to delete ScaleManager CR:
Command example:

```
oc -n ${FUSION_NS} delete scalemanager scalemanager
```

c. Run the following command to uninstall the IBM Storage Scale container native operator, Kubernetes objects, namespaces, and more:
Command example:

```
oc delete -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-
native/v5.2.0.x/generated/scale/install.yaml --ignore-not-found=true
```

Configuring storage

You can configure Global Data Platform as your storage.

- **Configuring Global Data Platform service on infrastructure nodes**
Manual steps to create the ScaleCluster CR and enable Global Data Platform service on infrastructure nodes.

Configuring Global Data Platform service on infrastructure nodes

Manual steps to create the ScaleCluster CR and enable Global Data Platform service on infrastructure nodes.

Before you begin

- Important: This procedure applies to configuring Global Data Platform service on Red Hat® OpenShift® Container Platform version 4.20 or earlier.

- You must have a minimum of two infrastructure nodes as two replicas of the `GUI` and `pmcollector` pods must run on different infrastructure nodes.
- Ensure that the infrastructure nodes have the following label:

  ```
  node-role.kubernetes.io/worker: ""
  ```

- The Global Data Platform service must be available.
- Important: Remote filesystem configuration must not exist.

Procedure

1. Apply the following YAML file to add IBM Storage Scale secrets for the remote mount.
   Update IBM Storage Scale secrets with the correct information.
   `ibm-spectrum-fusion-scale-csi-secret`:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: ibm-spectrum-fusion-scale-csi-secret-<remote_scale_clusterID>
     namespace: <fusion_namespace>
   type: kubernetes.io/basic-auth
   stringData:
     username: <scale_gui_user_name>
     password: <scale_gui_user_password>
   ```

   `ibm-spectrum-fusion-scale-gui-secret`:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: ibm-spectrum-fusion-scale-gui-secret-<remote_scale_clusterID>
     namespace: <fusion_namespace>
   type: kubernetes.io/basic-auth
   stringData:
     username: <scale_gui_user_name>
     password: <scale_gui_user_password>
   ```

   `ibm-spectrum-scale-csi-secret`:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: ibm-spectrum-scale-csi-secret-<remote_scale_clusterID>
     namespace: ibm-spectrum-scale-csi
   type: kubernetes.io/basic-auth
   stringData:
     username: <scale_gui_user_name>
     password: <scale_gui_user_password>
   ```

   `ibm-spectrum-scale-gui-secret`:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: ibm-spectrum-scale-gui-secret-<remote_scale_clusterID>
     namespace: ibm-spectrum-scale
   type: kubernetes.io/basic-auth
   stringData:
     username: <scale_gui_user_name>
     password: <scale_gui_user_password>
   ```

2. Add your filesystem details to the following YAML and create a `Scalecluster` CR:
   Update `Scalecluster` CR with the correct information.

   ```
   apiVersion: cns.isf.ibm.com/v1
   kind: ScaleCluster
   metadata:
     name: ibm-spectrum-fusion-scale-cluster-<remote_scale_clusterID>
     namespace: <fusion_namespace>
   spec:
     infraNode: true
     remoteScale:
       host: <remote_scale_gui_node_hostname_or_ip>
       clusterID: "<remote_scale_clusterID>"
       insecureSkipVerify: true
       contactNodes:
       - hostname: <contact_node1_hostname>
         ip: <contact_node1_ip>
       - hostname: <contact_node2_hostname>
         ip: <contact_node2_ip>
       fileSystems:
       - name: <remote_filesystem_name>
         scName: <custom_storageclass_name>
       guiSecret: ibm-spectrum-fusion-scale-gui-secret-<remote_scale_clusterID>
       csiSecret: ibm-spectrum-fusion-scale-csi-secret-<remote_scale_clusterID>
   ```

   If you enter a clusterID value without the double quotation marks in this example, then the `ibm-spectrum-fusion-scale-cluster-<remote_scale_clusterID>` CR (`Scalecluster` CR) creation fails with the following example error:

   ```
   Error from server: error when creating "ibm-spectrum-fusion-scale-cluster-xxx.yaml": admission webhook "mscalecluster.fusion.s
   ```

3. Apply the `Scalecluster` CR.
4. Monitor the filesystem and cluster status on the IBM Fusion user interface.

Setting up encryption for storage

In Global Data Platform storage, you can connect to an encrypted remote IBM Storage Scale file system. For Fusion Data Foundation storage, you can store the encryption key either as a secret in the cluster or in an external KMS.

- **Configuring encryption for Global Data Platform storage**
  You can configure the encryption in IBM Fusion to access an encrypted remote IBM Storage Scale file system.

Configuring encryption for Global Data Platform storage

You can configure the encryption in IBM Fusion to access an encrypted remote IBM Storage Scale file system.

Before you begin

Important: This procedure applies on Red Hat® OpenShift® Container Platform version 4.20 or earlier. On Red Hat OpenShift Container Platform version is 4.21 or later, encryption can be configured through the Red Hat OpenShift Container Platform web console when configuring the IBM Storage Scale cluster. For more information, see Fusion Data Foundation service with Global Data Platform.
Prepare IBM® Security Guardium® Key Lifecycle Manager (GKLM) server for IBM Fusion. To establish an encryption-enabled environment, see part 1 of Simplified setup: Using GKLM with a self-signed certificate.

Ensure that you go through the firewall recommendations for GKLM. See Firewall recommendations for IBM GKLM

For more information about encryption in IBM Storage Scale, see Encryption in IBM Storage Scale documentation.

About this task

The IBM Storage Scale admin encrypts the remote file system. As a IBM Fusion user, you must connect to the same key management server so that encrypted data can be accessed.

Procedure

1. Go to the Storage > Remote file systems page in IBM Fusion user interface to configure encryption for remote IBM Storage Scale filesystem.
2. Click Connect in the Encryption tile.
3. Enter the following connection details:

   Hostname
   : For local storage, enter the Security Key Lifecycle Manager host name to connect.

   Backup host name
   : Optionally, enter secondary GKLM server host name.

   Port number (optional)
   : The REST port number connects IBM Fusion to Security Key Lifecycle Manager REST admin interface. The default port number is 9443.
   Note: It can depend on Security Key Lifecycle Manager version. See Firewall recommendations for IBM GKLM.

   User name
   : The administrator user name for GKLM Server. The default value is `GKLMAdmin`.

   Password
   : The administrator password for the GKLM Server.

4. Enter the following Certificate details.
   Note: TLS/KMIP Certificates for secure communication on the KMIP port, only require when the key server is running with a certificate chain from a Certificate Authority (CA) rather than with a self-signed server certificate. The certificates must be formatted as PEM-encoded X.509 certificates.

   Root certificate
   : The root CA certificate from the Certificate Authority.

   Endpoint certificate
   : The server certificate that is signed by a CA.

   Intermediate certificate (optional)
   : The intermediate CA certificates are required only when the server certificate is signed by one of them. If you have more intermediate certificates, then click Add intermediate certificate to add them.

5. Enter the following values for File system tenants.

   Encryption tenant ID
   : Represents the keyspace configured on the GKLM server. All IBM Fusion systems that want to share or use encryption keys must use the same tenant ID.

   Remote key management ID
   : It is the remote key management ID. All nodes in the IBM Fusion system must use the same RKMID, which describes a combination of keyserver, tenant, and client on the remote scale cluster.

Important: The DNS name for the key server or tenant must start with a letter or number. Special characters at the beginning of the DNS name are not supported and can result in an invalid RKMID.

You can add more such Encryption tenant ID and Remote key management ID pairs.

Run the following commands on the remote scale cluster to retrieve the values:

```
mmkeyserv client show
```

This command gives the tenant name. If the tenant name is displayed as (none), then first register client using `mmkeyserv` client register command. For more information about this command, see mmkeyserv command.
To get Remote key management ID, run the following command on the remote scale cluster:

```
mmkeyserv tenant show
```

6. Click Configure.

Troubleshooting Global Data Platform issues

Global Data Platform (GDP) storage issues on Red Hat OpenShift Container Platform 4.20 or earlier can include filesystem mount delays, CSI driver errors, and backup failures with static persistent volumes.

Important: This section applies to troubleshooting issues related to the Global Data Platform service on Red Hat® OpenShift® Container Platform version 4.20 or earlier.
Warning: Do NOT delete IBM Storage Scale pods. Deletion of Scale pods in many circumstances has implications on availability and data integrity.

Filesystem status from Remote file systems is stuck in Connecting state

Cause

The secure boot is not supported by Scale and CNSA.

Resolution

1. Check whether the nodes are running with secure boot:

   ```
   mokutil --sb-state
   ```

   ```
   SecureBoot enabled
   ```

2. As a root password, you can disable the secure boot with the following command:

   ```
   mokutil --disable-validation
   ```

   ```
   password length: 8~16
   input password:
   input password again:
   ```

Delay between filesystem mount and IBM Storage Scale Container Storage Interface Driver (CSI) pods deployment

Diagnosis

CSI custom resource (CSO) status shows an error in getting the primary filesystem details. Run the following command to get the status:

```
oc describe cso -n ibm-spectrum-scale-csi
```

Example output:

```
Status:
  Conditions:
    Last Transition Time:  2023-06-06T07:10:08Z
    Message:               Failed to get the details of the primary filesystem: fs3
    Reason:                GetFileSystemFailed
    Status:                False
    Type:                  Success
```

Resolution

Delete only the CSI operator pod from `ibm-spectrum-scale-csi` namespace. For example, operator pod name `ibm-spectrum-scale-csi-operator-7bd7f4654d-28h56`.
Verify whether the CSO status show Success as True. For example:

```
Status:
  Conditions:
    Last Transition Time:  2023-06-06T07:11:16Z
    Message:               The CSI driver resources have been created/updated successfully
    Reason:                CSIConfigured
    Status:                True
    Type:                  Success
```

Backup of scale deployed application with static PV fails

Problem statement

This is a IBM Storage Scale limitation. Snapshots can be taken only for certain types of IBM Storage Scale volumes. Volume Snapshot is supported only for the independent fileset based PVCs. For more information about the limitation, see IBM Spectrum Scale documentation.

Resolution

As a resolution, reconfigure IBM Storage Scale and IBM Storage Scale Container Storage Interface Driver for independent fileset PVCs and try again. For steps to reconfigure, see IBM Spectrum Scale documentation.

Known issues

- If you deployed IBM Fusion to any custom namespace, the log collection package appears under the folder name ibm-spectrum-fusion-namespace. Instead, it must be under the custom namespace that is used in actual log collection.

Remote IBM Storage Scale storage cluster in IBM Fusion

IBM Storage Scale remote mount feature enables you to connect to a remote IBM Storage Scale cluster as though it is a part of your local cluster. The container native data access is only for IBM Fusion on On-premises, Bare Metal, VMware, and Amazon Web Services deployments.

Before you begin

- Important:
  Check your Red Hat® OpenShift® Container Platform version:
  - **Version 4.20 or earlier**: Continue with this procedure
  - **Version 4.21 or later**: see Fusion Data Foundation service with Global Data Platform
- For remote filesystem of Amazon Web Services storage cluster, ensure that you deploy a Amazon Web Services storage cluster. For more information about the storage, see AWS storage cluster.
- For the remote mounted storage cluster, after its filesystem level is updated to 28.00 (5.1.4) or later, it is recommended to enable `--auto-inode-limit` on its filesystem for IBM Fusion. For more information about how to enable, see mmchfs command.
- To access the container native data, you must first add a file system. To add a new remote file system connection, see Connecting to remote IBM Storage Scale file systems.
- To understand the number of file systems and clusters that can be remotely mounted, see IBM Spectrum Scale container native limitations.

About this task

- You can run a container workload against an existing data set that is stored on an IBM Storage Scale file system. For example, an IBM Storage Scale file system contains lot of images and you want to run a container workload that specializes in image processing.
- The remote file system mount must be configured. For more information, see Connecting to remote IBM Storage Scale file systems.
- When the remote file system mount is created, the IBM Fusion automatically creates a statically provisioned volume that points to the file system. You must know how to identify this Persistent Volume.
- Optionally, modify the file system path that is specified in the Persistent Volume so that it points to a specific directory in the file system, rather than the root.
- Now a Persistent Volume Claim (PVC) can be made against the Persistent Volume. It can be accomplished by matching the `accessMode` and storage request against the specified in the Persistent Volume.
- A workload that uses the Persistent Volume Claim can view the contents of the file system.
- Configure the UID/GID to match the one that is used to write the data set on the IBM Storage Scale file system. If you do not configure, then the container workload cannot access the files, because it does not have the permission.

Procedure

1. Click Storage > Remote file systems menu in the IBM Fusion user interface.
   It lists all existing connections to file systems with their details in a table format. The details include File System, FS status, Cluster, Cluster status, and Storage class. You can enter keywords in Search to search for a connection. You can also filter records based on the status.
2. Click ellipsis of a remote file system record to do the following actions:
   - Details- Click to view details about the remote file system connection.
   - Edit cluster- Click to change the cluster of a file system connection.
   - Remove- Click to delete a file system record.

Connecting to remote IBM Storage Scale file systems

In IBM Fusion, from a containerized workload, you can access and store data from existing IBM Storage Scale clusters.

Before you begin

Important:
Check your Red Hat® OpenShift® Container Platform version:

- **Version 4.20 or earlier**: Continue with this procedure
- **Version 4.21 or later**: see Fusion Data Foundation service with Global Data Platform

Firewall requirements

Important:

- For details of IBM Storage Scale container native storage access firewall requirements, see Network and firewall requirements.
- Grant Amazon Web Services Storage Cluster filesystem to be accessible for IBM Storage Fusion in ROSA environment, see AWS configuration.

Supported networking method

The supported networking method for IBM Storage Scale container native storage access is CNI plug-in for the core pods. As an admin, configure a suitable CNI plugin. For more information about configuration, see Container Network Interface (CNI) configuration.

- The IBM Fusion admin must exchange information with the IBM Storage Scale admin, and provide instructions on how the IBM Storage Scale admin must configure the file system sharing on the IBM Storage Scale side.
  - As a OpenShift Container Platform administrator, confirm whether you configured IBM Storage Scale remote file system mounts during the installation of IBM Fusion. You can check whether the IBM Fusion user interface displays a Remote file systems menu option in the left-navigation.
    Alternatively, run the following command to check whether the **Global Data Platform** value is set to true:

    ```
    oc describe spectrumfusion/spectrumfusion -n <fusion_namespace>
    ```

    The **oc** command is used wherever the Global Data Platform service is based on a legacy implementation. A sample output that confirms the **Global Data Platform** value:

    ```
    Spec:
      Data Protection:
        Enable:  false
      Global Data Platform:
        Enable:  true
      Open Shift Data Foundation:
        Auto Upgrade:  false
        Enable:        false
      License:
        Accept:  true
    ```

    If the value of the **Global Data Platform: Enable** is false, then enable it from the IBM Fusion user interface. For the procedure, see Installing Global Data Platform.
    To determine whether the appropriate remote mount service is installed, check the Services page.

  - **Important:** The IBM Storage Scale administrator must do a remote filesystem configuration on the IBM Storage Scale Container Storage Interface Driver storage cluster by following **steps 1** to **6** of Storage cluster configuration for Container Storage Interface (CSI) on non-AWS environment (like on-premises).
  - The IBM Storage Scale administrator in turn provides you the necessary information about cluster, file system, encryption algorithms. For instructions to share, click Instructions link in the Add IBM Storage Scale file system slide out pane. For the Instructions link, see Storage cluster.
    Note: The details include host name, port, Cluster ID, IBM Storage Scale certificate, user name, and password.
  - You can enter cluster access credentials information that is provided by the IBM Storage Scale admin into the IBM Fusion user interface, or place it into a CR by using YAML.
  - Run the following precheck commands on the IBM Storage Scale GUI node:
    - Check whether the user that does the remote mount configuration has correct access rights:

      ```
      /usr/lpp/mmfs/gui/cli/lsuser | grep ContainerOperator | grep CsiAdmin
      ```

      By default, the user passwords expire after 90 days. If the security policy of your organization permits it, use the **-e 1** option on the **mkuser** command to create a user with a password that does not expire. Also, create a unique GUI user for each OpenShift Container Platform cluster on the same remote Scale cluster.

      If not present, create the user with both ContainerOperator and CsiAdmin roles.

      ```
      /usr/lpp/mmfs/gui/cli/mkuser csi-storage-gui-user -p passw0rd -g CsiAdmin,ContainerOperator
      ```

      If you want to install CAS, then you need to assign the StorageAdmin role to the user created for remote mount.

      ```
      /usr/lpp/mmfs/gui/cli/chuser csi-storage-gui-user -g CsiAdmin,ContainerOperator,StorageAdmin
      ```

    - Run the following command to ensure all the nodes of your cluster are active:

      ```
      /usr/lpp/mmfs/bin/mmgetstate -a
      ```

      Sample output:

      ```
      Node number  Node name  GPFS state

                 1           host1      active

                 2           host2      active

                 3           host3      active
      ```

      If any of the node is down, analysis the issue and start the node.

    - Run the following command to validate whether the gpfsgui service is running:

      ```
      systemctl status  gpfsgui
      ```

      If gui service has issues, stop and start the service:

      ```
      systemctl stop gpfsgui.service
      systemctl start gpfsgui.service
      ```

      Log in to the GUI of **scalecluster** by using the validated username and password.

    - To get more information about the cluster, run **/usr/lpp/mmfs/bin/mmlscluster** command.

About this task

All the IBM Storage Scale commands are run by the IBM Storage Scale administrator on the remote IBM Storage Scale cluster.

`/usr/lpp/mmfs/bin/xxx`

Procedure

1. Click Storage > Remote file systems menu in the IBM Fusion user interface.
2. Click Add file system.
   The Add IBM Spectrum Scale file system slide out pane gets displayed.
3. Enter the following details to add one or more file systems.
   You can add one or more file systems of the existing cluster or choose to connect to a new cluster.
   Note: The OpenShift Container Platform admin needs the IBM Storage Scale admin to provide all of this information. The CLI commands referenced in this topic are run by the IBM Storage Scale admin on the remote IBM Storage Scale cluster.
   - Connect to a new cluster option:
     - Enter the following cluster access credentials for IBM Storage Scale:

       Host name
       : The host name is the GUI REST API endpoint. It needs to have a user with ContainerOperator and CsiAdmin roles. You can enter up to a maximum of three hosts.

       Port
       : The port is the GUI port, that is, the REST API endpoint port. By default, the port is 443. If the remote cluster uses other ports, you can provide it.

       Cluster ID
       : Enter the ID of the cluster. Contact your IBM Storage Scale administrator to get the cluster ID.

       IBM Spectrum Scale certificate
       : Enter the IBM Storage Scale root CA. If you provide the root CA value, the IBM Fusion verifies the certificate chain and hostname of the IBM Storage Scale cluster. If you do not provide the root CA value, the IBM Fusion skips the verification step.

       User name
       : Enter the GUI user name of the cluster.
       Note: Ensure that this user has both container operator permissions and IBM Storage Scale Container Storage Interface Driver administrator permissions.

       Password
       : Enter the password for the GUI user name.

     - Define up to three nodes on the IBM Storage Scale cluster to establish connection. These nodes are contact nodes, which is used to communicate with remote storage cluster. The projects and data get distributed between these nodes.
       Run the following command to get node details:

       `/usr/lpp/mmfs/bin/mmlscluster`

     - Enter the name and IP address for node 1, node 2, and node 3.
       Note: The name value must consist of lower case alphanumeric characters, '-' or '.', and must start and end with an alphanumeric character.
     - Enter the following details to identify the file system or system:
       - File system name - The name helps to identify the file system or systems that can be accessed through IBM Fusion. Run the following command to get the file system name:

         `/usr/lpp/mmfs/bin/mmlsconfig`

         The file system name must be an existing file system on the IBM Storage Scale cluster that you want to remote mount.

       - Storage class name - Enter a unique storage class name to add to this file system. The Storage class must not exist in the Red Hat OpenShift cluster.

       Note: Click Add to add more than one file system from this cluster. You can click the remove icon to delete a file system.
   - If you choose Select an existing connect cluster, then select the host name from the Select a cluster drop-down list. The cluster details are displayed, such as host name and name of the connected file system. Enter the following details to identify the file system:
     - File system name - The name helps to identify the file system or systems that can be accessed through IBM Fusion. Run the following command to get the file system name:

       `/usr/lpp/mmfs/bin/mmlsconfig`

       The file system name must be an existing file system on the IBM Storage Scale cluster that you want to remote mount.

     - Storage class name - Enter a unique storage class name to add to this file system. The Storage class must not exist in the Red Hat OpenShift cluster.

     Click Add to include more than one file system from this cluster. You can click the remove icon to delete a file system.
4. Click Add file system.
   An information message informs you that the file system is connecting. After the connection is complete, a success message appears to confirm that the file system is connected. The new connection gets added to the Remote file system list.

Backup & Restore

The Backup & Restore service protects your workloads with tailored backups by using a hub and spoke model. You can restore your data rapidly from local snapshots without having to rely on external storage. Backups can be transferred to external object storage solutions like IBM Cloud, Microsoft Azure, Amazon Web Services, and any S3-compliant object storage. It allows the protection of the IBM Fusion Backup & Restore service itself to a S3 bucket.

- **Backup & Restore overview**
  You can protect your container workloads and virtual machines by using the IBM Fusion Backup & Restore service.
- **Install Backup & Restore**
  Protect your data with application-centric backups. Use local snapshots for quick recovery or transfer backups to external object storage for safe keeping. The IBM Fusion Backup & Restore service installation include backup hub and backup spoke.
- **Uninstalling Backup & Restore**
  Steps to uninstall IBM Fusion Backup & Restore by using command-line or command prompt interface.
- **Getting started**
  To quick start the service, define the StorageClass and make sure the VolumeSnapshotClass matches the StorageClass provisioner. Next, connect your clusters and set up the backup locations. Create and assign backup policies to the relevant applications, and regularly back up for continuity. This section also introduces you to the protection of the Backup & Restore service itself from data corruption and loss.
- **Backup & Restore of your workloads**
  The Backup & Restore service is designed to protect a wide range of workloads, including virtual machines, Kubernetes applications, AI workloads, and various IBM Cloud Paks. Additionally, it can also back up the Backup & Restore data management service to an S3 bucket.
- **Managing backup for multi-cluster environment**
  You can connect and manage cluster connections in IBM Fusion. It includes steps to establish connections and manage existing connections. You can also reconnect an OpenShift Container Platform cluster either after a disaster recovery event or a temporary unavailability.
- **Backup consistencies**
- **Performance and scaling**
  Backup and restore performance depends on agent concurrency, data mover pod counts, and storage throughput. Tuning these parameters helps IBM Fusion meet capacity and latency requirements for large workloads.
- **FIPS-140-2**
  The Federal Information Processing Standard Publication 140-2 (FIPS-140-2) is a standard that defines a set of security requirements for the use of cryptographic modules. Law mandates this standard for the US government agencies and contractors and is also referenced in other international and industry specific standards.
- **Backup & Restore service installation and upgrade issues**
  Use this troubleshooting information to resolve install and upgrade problems that are related to Backup & Restore service.
- **Troubleshooting Backup & Restore service issues**
  List of known Backup & Restore issues in IBM Fusion.

Backup & Restore overview

You can protect your container workloads and virtual machines by using the IBM Fusion Backup & Restore service.

In the event of data loss or corruption, such as accidental deletion or malicious ransomware attacks, a backup and restore system is needed to ensure data integrity and availability. By taking periodic backups of applications and storing them safely, you can restore your workloads during disasters, thereby minimizing downtime and data loss. Whenever disaster occurs, you can restore the workloads in place or to an alternate cluster.

Backup & Restore service protects both containerized and virtual machine (VM) workloads, going beyond data protection to include the application's state and essential etcd resources. This capability is not only for stateful applications but also for stateless ones, ensuring complete restoration of functionality and configuration.

Backup and restore works by first choosing a secure backup location and then defining backup polices tailored to your application needs with frequency and retention parameters. You can then associate the workloads with defined backup policies to automate and streamline the backup process. Finally, you can recover the backed up workloads either in their original location or to an alternate cluster based on the requirements.

For more complex scenarios or whenever you need to run tailored and repeatable workflows, you can make use of the advanced recipes provided by Backup & Restore service.

Backup & Restore supports both single-cluster mode and hub and spoke model. In a single cluster mode, all backup and restore operations are managed within a single cluster. In case of hub and spoke model, the hub acts as a central management point and spokes are individual clusters that run backup and restore tasks locally. You can choose this mode to manage backups across multiple clusters, and it is useful for large-scale environments where multiple clusters need to be managed with resilience in case of failures.

The IBM Fusion Backup & Restore service protection involves the backup of the Backup & Restore data management service itself to a S3 bucket.

- **Backup and restore planning**
  Plan resource requirements for backups.
- **How Backup & Restore service backups up application data**
  Backup & Restore service includes support for block volume mode PVs along with the backup of applications with filesystem volume mode PVs. It also supports incremental backups for stateful data (PVC) using Fusion Backup & Restore.
- **Backup & Restore service architecture**
  This topic explains the high-level workings of the Backup & Restore service to replicate data. It provides an overall idea of the internal components and their roles in the backup and restore operation, resulting in the final transfer of data to designated storage.
- **Hub and spoke model overview**
  Hub and spoke model allows a single administrator to manage backups for multiple applications that exist on the different clusters.

Backup and restore planning

Plan resource requirements for backups.

The first backup is a full backup. All subsequent backups are incremental.

The capacity required for a backup repository on a remote S3 object store depends on the initial size of all application PVCs and resources, the application data change rate, the backup frequency, and the retention settings. Because the repository is remote, the network must support transferring data volumes equivalent to the S3 object store size.

For example, consider an application that uses 2 TB of data and generates 300 MB of changed data per day. With a daily backup policy and 7-day retention, the initial backup transfers and stores 2 TB of data. Each subsequent daily incremental backup transfers and stores 300 MB.

IBM Fusion supports backup and recovery operations on Red Hat® OpenShift® Container Platform environment with any storage provider that implemented the Container Storage Interface driver and meets the following prerequisites:

- The Container Storage Interface driver must be v1 or higher (no support for *alpha* or *beta* versions)
- The Container Storage Interface driver must support Volume Snapshot and Restore.

Note: It is also recommended that the `VolumeSnapshotClass` deletion policy is configured to delete snapshots after the corresponding `VolumeSnapshotContent` is deleted (`deletionPolicy: Delete`).

How Backup & Restore service backups up application data

Backup & Restore service includes support for block volume mode PVs along with the backup of applications with filesystem volume mode PVs. It also supports incremental backups for stateful data (PVC) using Fusion Backup & Restore.

You can backup an application data, and it consists of PVC data and resources. For the PVC data, you can use online backups or snapshots. You mount the snapshot as a separate PVC for independent processing of the backup without impacting the application. The first back is a full backup and all subsequent backups are incremental. The backups are store in a S3 repository or locally on the system using snapshots (not recommended). The detection and processing of Block volume mode PVs is automatic with no user configuration or changes to the input.

Backup and restore of PV

The Backup & Restore service can backup data hosted on both `volumeMode: Block` and `Filesystem persistent volumes (PV)`. The detection and processing of the `volumeMode` is automatic and does not need any user configuration or changes.

PV with volumeMode: Filesystem
: The pod mounts it as a filesystem directory.

PV with volumeMode: Block
: The pod mounts it as a raw block device without any filesystem on it. This mode provides the best performance as it eliminates the filesystem layer between the pod and the PV. A common application for block PVs is database and virtual machines, that is, Red Hat® OpenShift® Virtualization.

Example Block volume mode PersistentVolume definition.

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: local-pv-76de1a29
  uid: 42643cc7-ac33-4536-b31b-6774524ee590
spec:
  capacity:
    storage: 3576Gi
  local:
    path: /mnt/local-storage/mylocalvs/nvme-Dell_Ent_NVMe_CM6_RI_3.84TB_Y2B0A01HTCE8
  accessModes:
    - ReadWriteOnce
  claimRef:
    kind: PersistentVolumeClaim
    namespace: openshift-storage
    name: ocs-deviceset-mylocalvs-0-data-4xh44d
    uid: 9abba1a3-6cd9-4d61-9f3e-95b29dab2094
    apiVersion: v1
    resourceVersion: '71161'
  persistentVolumeReclaimPolicy: Delete
  storageClassName: mylocalvs
  volumeMode: Block
```

Backup and restore of PVC data

Stateful data (PVC) backups with Fusion Backup & Restore are incremental, meaning that only new or changed data since the previous backup is sent to the backup storage location. After the initial backup, which copies all stateful data on the PVCs, subsequent backups only transfer new or modified data. After the first backup, the entire data is never copied again; instead, backups continue to send only new or changed data.

RADOS Block Device (RBD) storage class based volumes

In RBD storage class based volumes, change block detection capability speeds up processing time of the backup.

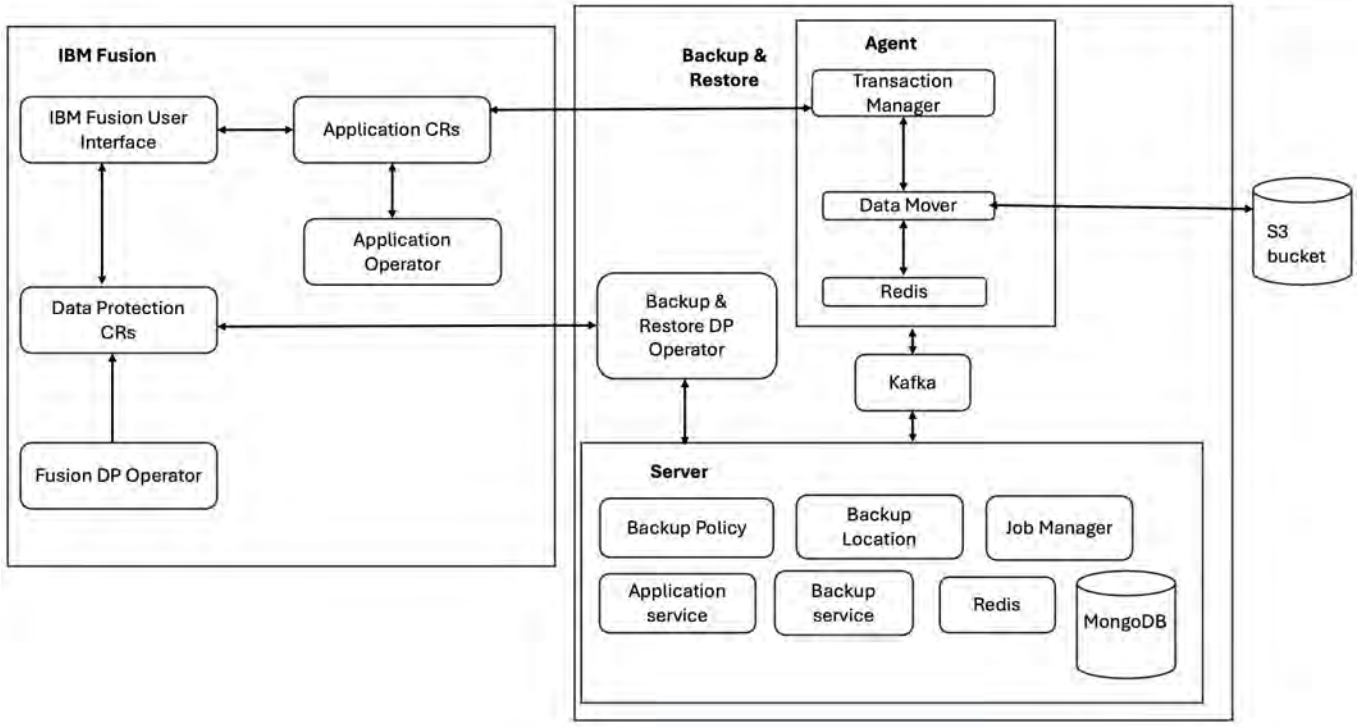
Resources

Kubernetes resource data is always copied in its entirety on each backup operation. Comparatively, the Kubernetes resource data represents a very small percentage of the total amount of data that gets transferred for the backup operation.

Backup & Restore service architecture

This topic explains the high-level workings of the Backup & Restore service to replicate data. It provides an overall idea of the internal components and their roles in the backup and restore operation, resulting in the final transfer of data to designated storage.

The following diagram depicts the workflow of a Backup & Restore hub and local agent.



The IBM Fusion CRs (Application and Data Protections CRs) and operators (Application and Fusion DP operator) interact with the Backup & Restore service. The Backup & Restore service includes Agent and Server components.

Agents handle specific tasks to perform up to ten backup or restore jobs at once:

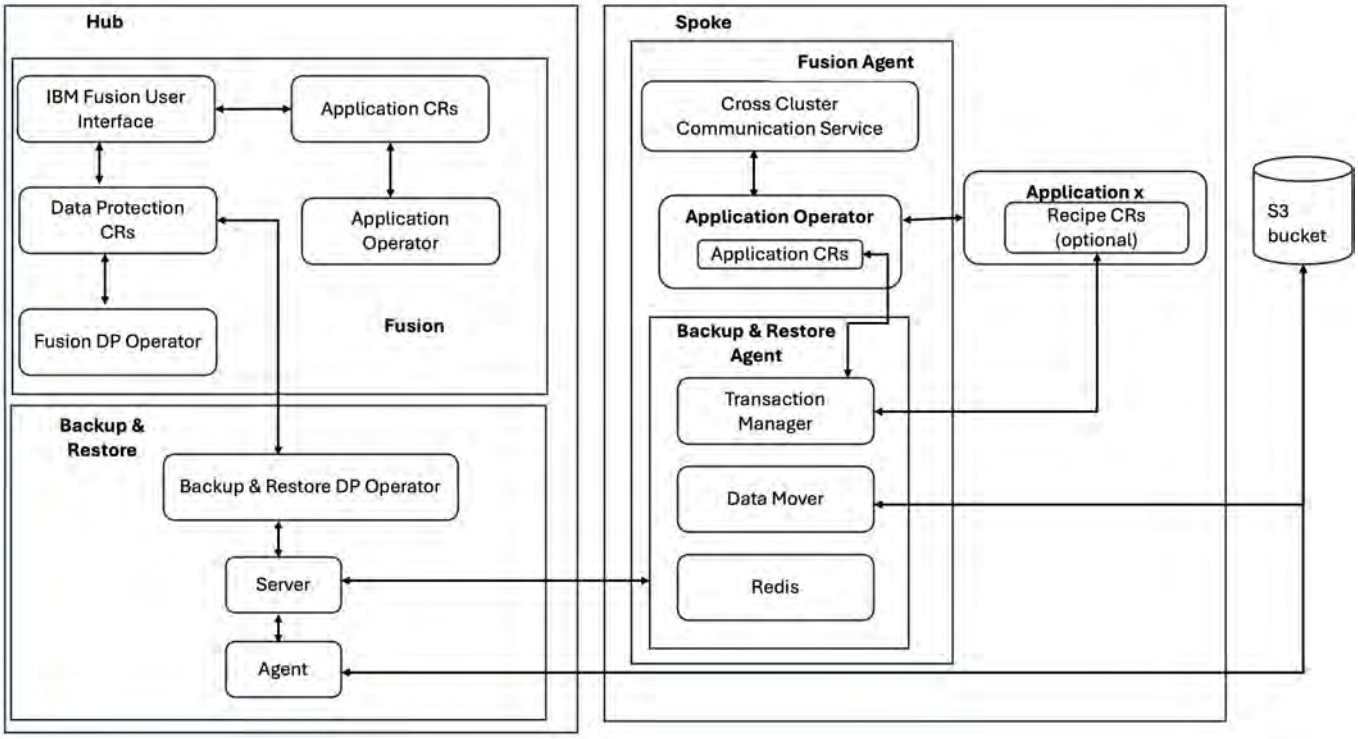
- The Transaction Manager facilitates the backup or restore job requests from the server. These actions include request for PVC snapshots, recipe processing, and status updates to the server.
- To process the backup, the Data Mover mounts the PVC snapshot to read the data, determines any deduplication, and then transfers the data to the designated S3 bucket.

The server includes components like Backup Policy, Backup Location, Job Manager, Application Service, Backup Service, Redis, and MongoDB.

- The Backup Policy manages scheduling and cleanup tasks through Policy CRs.
- the Backup Location validates BSL CRs without performing extensive validations as it directly communicates with the S3 bucket to verify connectivity and existence.
- The Job Manager distributes jobs to agents and monitors job progress to detect and cancel hung jobs. It also prevents duplicate jobs for the same application or policy combination from running at the same time.
- Applications Service maintains records of all existing applications across clusters.
- The Backup Service updates backup and restore Custom Resource with their respective statuses.
- Redis is employed to manage lock contentions; for instance, during scheduled backups involving replica instances, after one acquires a Redis lock, it initiates the task while others are dormant.
- MongoDB functions as the central repository for backup and metadata details.

Backup & Restore service uses Kafka as the primary communication medium among services, providing scalability and resilience. It also allows multiple replicas to process distinct messages concurrently.

The following diagram depicts the workflow of a Backup & Restore hub with multiple spokes.



In case of hub and spoke, it is similar but the agents are on different clusters. The Fusion UI, CRs, Backup & Restore server and local agent are the same except that it needs additional cross cluster communication.

When you attach a spoke cluster to the hub, install the Backup & Restore Agent service on the spoke. The installation of the Backup & Restore Agent service requires a temporary token provided by the Hub for initial setup of the secure cross-cluster communication between the hub and spoke.

Hub and spoke model overview

Hub and spoke model allows a single administrator to manage backups for multiple applications that exist on the different clusters.

The Hub is the single location from where the administrator can login and manage all the applications on all the spoke and hub clusters. The hub location hosts the server and facilitates the Backup & restore administrator to run backup and restore jobs across all clusters.

The hub includes all the Backup & restore server components and agent components for backing up components on its own cluster. The spoke has only the Backup & restore agent installed in it. The Backup & restore hub maintains all scheduling, retention, policy handling, location management, and everything else from a management point of view. The agent or spoke takes care of requests to facilitate Backup & restore jobs on that cluster.

Whenever you want to manage applications on a spoke cluster, you must install IBM Fusion and the Backup & restore Agent Service on that cluster. As part of the spoke agent installation process, a communication is established between the hub and spoke cluster. The spoke Agent installation requests for a connection snippet to be generated on the hub and you must provide it within an hour to the Agent installation. This temporary connection snippet generates a permanent communication between the hub and spoke by generating a certificate from the Certificate Authority (CA) of the hub. By default, this certificate expires after 90 days. IBM Fusion takes care of the needed steps to auto-renew a certificate before it expires.

IBM Fusion is also resilient to unreliable network connections and can recover with network disconnections between the hub and spoke clusters up to an hour .

In the Jobs page of the hub cluster, you can view all the backup and restore jobs across all clusters. You can filter jobs of a specific cluster. When a backup or restore job gets initiated, it gets remotely deployed to the cluster where the agent picks up the job to run it locally on that cluster.

The Service Protection feature provides the ability to restore applications from remote S3 backups (local snapshot only backups not supported) across all hub and spoke clusters. While the Service Protection backup is configured on the hub cluster, all backup information for spoke clusters is automatically included and no configuration on the spoke clusters are required.

In the event of a cluster crash, you can restore application backups on the replacement cluster.

Note: Service protection is just for backup restore on the hub cluster and not for other configurations that exist in IBM Fusion. For example, Red Hat® OpenShift® Container Platform cluster, disaster recovery, Red Hat OpenShift Data Foundation.
In the Topology page, you can see the hub cluster in the first row followed by all connected spoke clusters. The details of the connected clusters, such as health, status, success rate are available in this view.

The two available backup storage locations are the in place local snapshots and S3 remote object store locations. If you want to restore applications on another cluster, configure a S3 remote object store for the backup storage locations. The S3 remote object store is to store data on an external location. During instances wherein the cluster crashes, the backups that are stored in the local snapshots are lost along with it. Hence, it is important to offload your backups to an external storage location, and later restore the application on a replacement cluster.

You can share the locations and policies for applications that are backedup up across the cluster. From the hub cluster, you can apply policies to any number of applications across any number of clusters. All applications get backed up at the same time as they share policies and the schedules. However, it can cause resource contention in the network cluster. For example, all backups need network connection to a S3 location at the same time. Though it gives the advantage of maintaining limited number of policies, it might place a strain on the network. Similar to the policies, you can use one location to store every backup of all clusters. It can cause contention of network requests on the same backup location.

The Applications page shows only applications on that cluster and not any applications on the spoke cluster. The Backedup application page shows all applications that are protected across the cluster. To protect new applications in the cluster, click Protect apps in the Backedup application page, and select applications from a specific cluster.

Install Backup & Restore

Protect your data with application-centric backups. Use local snapshots for quick recovery or transfer backups to external object storage for safe keeping. The IBM Fusion Backup & Restore service installation include backup hub and backup spoke.

About this task

For more information about the hub and spoke architecture of Backup & Restore service, see Hub and spoke model overview.

- **Prerequisites**
  Ensure that you meet the prerequisites before you work with Backup & Restore service.
- **Backup & Restore hub**
  Protect your data with application-centric backups. Use local snapshots for quick recovery or transfer backups to external object storage for safe keeping.
- **Backup & Restore spoke**
  Protect your data with application-centric backups. Use local snapshots for quick recovery or transfer backups to external object storage for safe keeping.
- **Establishing connection between hub and spoke**
  Generate YAML from the hub cluster to establish mutual authentication between the hub and spoke clusters.

Prerequisites

Ensure that you meet the prerequisites before you work with Backup & Restore service.

1. Install either the Backup & Restore or Backup & Restore Agent service. For the procedure to install, see Install Backup & Restore.
2. Ensure that you define a StorageClass that is provisioned by a supported CSI driver. Check the `provisioner:` in the `Spec` to make sure that it is backed by CSI.
   Sample Storage Class for Fusion Data Foundation:

   ```
   apiVersion: storage.k8s.io/v1
   kind: StorageClass
   metadata:
     name: ocs-storagecluster-cephfs
   provisioner: openshift-storage.cephfs.csi.ceph.com
   reclaimPolicy: Delete
   allowVolumeExpansion: true
   volumeBindingMode: Immediate
   ```

   Note: The Storage Class must have the CSI Snapshot capability (`VolumeSnapshotClass`).
3. For backups and restores to run successfully, ensure that CSI compliant storage classes have `volumeBindingMode: Immediate` or `volumeBindingMode: WaitForFirstConsumer` and `allowVolumeExpansion: true`.
4. Ensure that the VolumeSnapshotClass has a `driver:` that matches the `provisioner:` in the StorageClass.
   Sample YAML for Red Hat® OpenShift® Data Foundation RBD storage:

   ```
   apiVersion: snapshot.storage.k8s.io/v1
   kind: VolumeSnapshotClass
   driver: openshift-storage.cephfs.csi.ceph.com
   metadata:
     name: ocs-storagecluster-cephfsplugin-snapclass
   deletionPolicy: Delete
   parameters:
     clusterID: openshift-storage
     csi.storage.k8s.io/snapshotter-secret-name: rook-csi-cephfs-provisioner
     csi.storage.k8s.io/snapshotter-secret-namespace: openshift-storage
   ```

Backup & Restore hub

Protect your data with application-centric backups. Use local snapshots for quick recovery or transfer backups to external object storage for safe keeping.

Before you begin

Consider the following points before you begin installation:

- Support for Backup & Restore service on Amazon web services ROSA is using native storage class - gp2-csi, gp3, and gp3-csi. The native storage classes gp2-csi, gp3 and gp3-csi are supported if you create a new class based on those that has volumeBindingMode Immediate. The gp3 and gp3-csi classes are clones of each other, one can be used as a base to create a new class with immediate binding.
- Support for Backup & Restore service on Microsoft Azure ARO is using managed-csi storage class. The managed-csi storage class is CSI compliant, but does not have volumeBindingMode Immediate. You can create a new storage class based on managed-csi and set Immediate binding mode.
- Firewall ports required for Hub and Spoke architecture:
  - Hub
    Must be able to make a TCP connection to the Spoke cluster API address

- A route on the host that creates a DNS address exists for the Kubernetes API, which is enabled by default during the installation of Red Hat® OpenShift®. Check whether it is available and is resolvable from the spoke containers. Format of the URL is `api.<cluster-name>.<domain>` but is changeable. This is port 443 on all control plane nodes.
- A route to the Kafka Bridge creates a DNS address. Check whether it is available and is resolvable from the spoke containers. Uses port 443 on compute nodes. Run the following command and check the role in the output to know which nodes are compute nodes (needed for Kafka Bridge) and control plane nodes (for Kubernetes API connection):

  ```
  oc get nodes
  ```

  Example output:

  ```
  NAME STATUS ROLES AGE VERSION
  bootstrap.ocpfsn.pok.stglabs.ibm.com Ready worker 3d18h v1.25.14+20cda61
  master0.ocpfsn.pok.stglabs.ibm.com Ready control-plane,master 3d19h v1.25.14+20cda61
  master1.ocpfsn.pok.stglabs.ibm.com Ready control-plane,master 3d19h v1.25.14+20cda61
  master2.ocpfsn.pok.stglabs.ibm.com Ready control-plane,master 3d19h v1.25.14+20cda61
  worker0.ocpfsn.pok.stglabs.ibm.com Ready worker 3d18h v1.25.14+20cda61
  worker1.ocpfsn.pok.stglabs.ibm.com Ready worker 3d18h v1.25.14+20cda61
  worker3.ocpfsn.pok.stglabs.ibm.com Ready worker 2d2h v1.25.14+20cda61
  ```

  Alternatively, to check the roles from OpenShift console, do the following steps:
  1. Log in to the OpenShift console.
  2. Click Compute > Nodes menu.
  3. Check the role of the nodes.
- The following command can be used to get the cluster API address of a cluster:

  ```
  oc cluster-info
  ```

  For example:

  ```
  Kubernetes control plane is running at https://c109-e.us-east.containers.cloud.ibm.com:30363
  ```

- Backup & Restore hub can be deployed as containers on Red Hat OpenShift Virtualization Engine (OVE) but must run on normal worker nodes.

Procedure

1. Go to Services page in IBM Fusion user interface.
2. In the Available section, click the Backup & Restore tile.
3. In the Backup & Restore page, go through the features and capabilities of the service and click Install.
4. In the Install service window, select a Storage class that is used for the service.
   Select `ocs-storagecluster-ceph-rbd`for Backup & Restore service. The internal data catalog requires a minimum of 200 GB for ReadWriteAfter storage.
5. Click Install.
   The installation starts and a notification appears on the Services page. You can see the progress of the installation in the Services > Installed section. After the installation completes successfully, you can see the status as normal and a Get started link.
   After you enable the Backup & Restore, you can view the service version and health status. From the ellipsis menu, you can download logs and view documentation. After you successfully collect the logs, a success notification gets displayed. The notification disappears automatically after some time.

   The Backup & Restore menu in the spoke cluster includes the following sub-menus:
   - Topology
   - Backed up applications
   - Policies
   - Locations
   - Jobs
   - Service protection

   In the Backup & restore > Overview page, you have quick links to generate YAML, Connect locations, Create backup policies, and protect your applications.

   In case of other failures, go through the downloaded logs to understand the cause of the failure and fix the issue. For more information about service issues in IBM Fusion, see Backup & Restore service installation and upgrade issues.
6. Validate the installation:
   Note: Ensure that the Backup & Restore IBM Fusion service is enabled before validation.

   Verify that the Backup & Restore IBM Fusion HCI service operators from the OpenShift Container Platform web console:
   a. Go to Installed operators from OpenShift Container Platform web console.
   b. Select the project as **`ibm-backup-restore`**.
   c. Verify that the following operators show the status as **`succeeded`**.
      - Red Hat Integration - AMQ Streams
      - IBM Fusion Backup & Restore Hub
      - IBM Fusion Backup & Restore Spoke
      - OADP Operator
   d. Alternatively, you can verify the status of operators by running the following oc command:
      Note: More operators appear in the command line output than in the web console.

      ```
      oc get csv -n ibm-backup-restore
      ```

      A sample result of the oc command is as follows:

      ```
      NAME                              DISPLAY                                       VERSION     REPLACES                PHASE
      amqstreams.v2.3.0-1               Red Hat Integration - AMQ Streams             2.3.0-1   amqstreams.v2.3.0-0
      Succeeded
      guardian-dm-operator.v2.5.0       IBM Fusion Backup and Restore Data Mover      2.5.0
      Succeeded
      guardian-dp-operator.v2.5.0       IBM Fusion Backup and Restore Data Protection 2.5.0
      Succeeded
      guardian-mongo-operator.v2.5.0    IBM Fusion Backup and Restore Mongo           2.5.0
      Succeeded
      ibm-dataprotectionagent.v2.5.0    IBM Fusion Backup and Restore Agent           2.5.0
      ```

```
Succeeded
ibm-dataprotectionserver.v2.5.0   IBM Fusion Backup and Restore Server            2.5.0
Succeeded
oadp-operator.v1.1.2              OADP Operator                                   1.1.2     oadp-operator.v1.1.1
Succeeded
redis-operator.v2.5.0             IBM Fusion Backup and Restore Redis            2.5.0
Succeeded
```

.Ensure that the status shows **Succeeded**.

Verify the Backup & Restore pods from the OpenShift Container Platform console:

a. Go to Workloads > Pods.

b. Select the namespace, where you installed IBM Fusion Backup & Restore. In this case, select **ibm-backup-restore** namespace.
It lists all the pods. Ensure that all pods are running.

c. Verify whether the following pods are running successfully:

```
amq-streams-cluster-operator
applicationsvc
backup-location-deployment
backup-service
backuppolicy-deployment
guardian-dm-controller-manager
guardian-dp-operator-controller-manager
guardian-kafka-cluster-entity-operator
guardian-kafka-cluster-kafka-0
guardian-kafka-cluster-kafka-1
guardian-kafka-cluster-kafka-2
guardian-kafka-cluster-zookeeper-0
guardian-kafka-cluster-zookeeper-1
guardian-kafka-cluster-zookeeper-2
guardian-minio-0
guardian-mongo-operator-controller-manager
ibm-dataprotectionagent-controller-manager
ibm-dataprotectionserver-catalog-ibm-backup-restore
ibm-dataprotectionserver-controller-manager
ibm-backup-restoreagent-controller-manager
ibm-backup-restoreserver-controller-manager
job-manager
mongodb-0
mongodb-1
mongodb-2
mongodb-ab-0
openshift-adp-controller-manager
redis-master-0
redis-operator-controller-manager
redis-replicas-0
redis-replicas-1
redis-replicas-2
transaction-manager
velero
```

d. Alternatively, you can verify the installation by running the following oc command:

```
oc get pods -n ibm-backup-restore
```

A sample result of the oc command is as follows:

```
NAME                                                              READY   STATUS      R
amq-streams-cluster-operator-v2.3.0-1-7d6fb79d84-jdkfh            1/1     Running     0
applicationsvc-55c9b4d6c9-6hdv7                                   1/1     Running     0
b0f64f9161e0882f278dde2eaa1ea9677f4a230a29180fcf21fc665761hvvxz   0/1     Completed   0
backup-location-deployment-6b565b856c-j4vjc                       1/1     Running     0
backup-service-54bf9988f6-47bpv                                   1/1     Running     0
backuppolicy-deployment-b997cc9bf-dfwnh                           1/1     Running     0
bc6176f08ef686ccde24395724b77ea07a586a0bd1fa27ebfd5d704d0dxv9pl   0/1     Completed   0
e349f7c16f02ad6c0c31e41ba2fb1750d5154b58537224d239fe47508872cj5   0/1     Completed   0
f3dd0cbe8cb98614cf163fc5372733148236ea2bdeb5efc6a5d5afe4c085qlk   0/1     Completed   0
ff53c6d827e0fd610a4392c4f9411beb0c785572d2fca1bda57e208650bwv2m   0/1     Completed   0
ffacc5cdaa1e0aa4f3b3c0021f3c87931aa7422f8415303d95457febc8nmzk7   0/1     Completed   0
guardian-dm-controller-manager-64d57bf9ff-28dqj                   2/2     Running     0
guardian-dp-operator-controller-manager-6f6d55f6f7-fhndb          2/2     Running     0
guardian-kafka-cluster-entity-operator-b59d699f7-5qxt8            3/3     Running     0
guardian-kafka-cluster-kafka-0                                    1/1     Running     0
guardian-kafka-cluster-kafka-1                                    1/1     Running     0
guardian-kafka-cluster-kafka-2                                    1/1     Running     0
guardian-kafka-cluster-zookeeper-0                                1/1     Running     0
guardian-kafka-cluster-zookeeper-1                                1/1     Running     0
guardian-kafka-cluster-zookeeper-2                                1/1     Running     0
guardian-minio-0                                                  1/1     Running     0
guardian-mongo-operator-controller-manager-6f47776cb4-s6tkm       2/2     Running     0
ibm-dataprotectionagent-controller-manager-54d66f7975-lgdgh       2/2     Running     0
ibm-dataprotectionserver-catalog-ibm-backup-restore-k967t         1/1     Running     0
ibm-dataprotectionserver-controller-manager-749554d89f-q6gmx      2/2     Running     0
job-manager-859484bfc5-fzpt8                                      1/1     Running     0
mongodb-0                                                         2/2     Running     1
mongodb-1                                                         2/2     Running     1
mongodb-2                                                         2/2     Running     1
mongodb-ab-0                                                      1/1     Running     0
openshift-adp-controller-manager-59fb9f86f4-gdbhb                 1/1     Running     0
redis-master-0                                                    1/1     Running     0
redis-operator-controller-manager-647cf89ff7-wl8w2                2/2     Running     0
redis-replicas-0                                                  1/1     Running     0
redis-replicas-1                                                  1/1     Running     0
redis-replicas-2                                                  1/1     Running     0
```

```
transaction-manager-5d9b59cf9f-hdzjm                              2/2     Running     0
velero-777d65c9b7-455fv                                           1/1     Running     0
```

Verify the Backup & Restore IBM Fusion service installation status from the OpenShift Container Platform web console:

a. Go to Installed operators from OpenShift Container Platform web console.
b. Select the namespace as `ibm-backup-restore`.
c. Select IBM Fusion Backup & Restore server.
d. Click the Data Protection server tab and select Data Protection server.
e. Select the YAML tab.
f. In the status section, makes sure the following:
   - **`HealthStatuses`** Shows the health status of all components listed. It may take 5 minutes or more for all components to show up as healthy.
   - Ensure the install status shows Complete and **`progressPercentage`** as 100.
g. Alternatively, you can verify the installation status by running the following oc command:

```
oc describe dataprotectionserver dataprotectionserver -n ibm-backup-restore
```

A sample result of the oc command is as follows:

```
Status:
  Health Statuses:
    Service Name:  applicationservice
    Status:        Healthy
    Service Name:  backuplocation
    Status:        Healthy
    Service Name:  backuppolicy
    Status:        Healthy
    Service Name:  backupservice
    Status:        Healthy
    Service Name:  jobmanager
    Status:        Healthy
    Service Name:  backupagent
    Status:        Healthy
    Service Name:  mongo
    Status:        Healthy
    Service Name:  redis
    Status:        Healthy
    Service Name:  kafka
    Status:        Healthy
  Install Status:
    Progress Percentage:  100
    Retry On Failure:     false
    Status:               Completed
  Installed Version:      2.5.0
  Upgrade In Progress:    false
  Upgrade Status:
    Retry On Failure:  false
```

h. Check the status section at the end of the CR and make sure the following:
   - The Install Status section shows the status of the install along with the percentage complete.
   - The install status shows the install status as Completed when the installation is successfully completed.
   - The health status section lists and shows components health status as healthy.
   - Note: Individual component health statuses may show **`Unknown`** or **`Degraded`** for up to five minutes, and show a healthy status when installation is complete.

What to do next

- Generate a connection snippet or YAML. For the procedure to generate, see Establishing connection between hub and spoke.
- You can now begin to protect your IBM Fusion applications.
  1. Add location to determine whether the network verification is needed. For the procedure to add a location, see Adding backup storage location.
  2. Creating backup policy.
  3. Assigning backup policy.

Backup & Restore spoke

Protect your data with application-centric backups. Use local snapshots for quick recovery or transfer backups to external object storage for safe keeping.

Before you begin

- Install backup hub. For the procedure to install, see Backup & Restore hub.
- Generate the YAML. This YAMLis required to establish mutual authentication between the two clusters. For the procedure to generate, see Establishing connection between hub and spoke.
- When you add a Spoke to a Hub, the version of the Spoke must be the same version as the Hub.
- Consider the following points before you begin installation:
  - Support for Backup & Restore service on Amazon web services ROSA is using native storage class - gp2-csi, gp3, and gp3-csi. The native storage classes gp2-csi, gp3 and gp3-csi are supported if you create a new class based on those that has volumeBindingMode Immediate. The gp3 and gp3-csi classes are clones of each other, one can be used as a base to create a new class with immediate binding.
  - Support for Backup & Restore service on Microsoft Azure ARO is using managed-csi storage class. The managed-csi storage class is CSI compliant, but does not have volumeBindingMode Immediate. You can create a new storage class based on managed-csi and set Immediate binding mode.
  - Firewall ports required for Hub and Spoke architecture:
    - Hub
      Must be able to make a TCP connection to the Spoke cluster API address

- A route on the host that creates a DNS address exists for the Kubernetes API, which is enabled by default during the installation of Red Hat® OpenShift®. Check whether it is available and is resolvable from the spoke containers. Format of the URL is `api.<cluster-name>.<domain>` but is changeable. This is port 443 on all control plane nodes.
- A route to the Kafka Bridge creates a DNS address. Check whether it is available and is resolvable from the spoke containers. Uses port 443 on compute nodes. Run the following command and check the role in the output to know which nodes are compute nodes (needed for Kafka Bridge) and control plane nodes (for Kubernetes API connection):

  ```
  oc get nodes
  ```

  Example output:

  ```
  NAME STATUS ROLES AGE VERSION
  bootstrap.ocpfsn.pok.stglabs.ibm.com Ready worker 3d18h v1.25.14+20cda61
  master0.ocpfsn.pok.stglabs.ibm.com Ready control-plane,master 3d19h v1.25.14+20cda61
  master1.ocpfsn.pok.stglabs.ibm.com Ready control-plane,master 3d19h v1.25.14+20cda61
  master2.ocpfsn.pok.stglabs.ibm.com Ready control-plane,master 3d19h v1.25.14+20cda61
  worker0.ocpfsn.pok.stglabs.ibm.com Ready worker 3d18h v1.25.14+20cda61
  worker1.ocpfsn.pok.stglabs.ibm.com Ready worker 3d18h v1.25.14+20cda61
  worker3.ocpfsn.pok.stglabs.ibm.com Ready worker 2d2h v1.25.14+20cda61
  ```

  Alternatively, to check the roles from OpenShift console, do the following steps:
  1. Log in to the OpenShift console.
  2. Click Compute > Nodes menu.
  3. Check the role of the nodes.
- The following command can be used to get the cluster API address of a cluster:

  ```
  oc cluster-info
  ```

  For example:

  ```
  Kubernetes control plane is running at https://c109-e.us-east.containers.cloud.ibm.com:30363
  ```

- Backup & Restore hub can be deployed as containers on Red Hat OpenShift Virtualization Engine (OVE) but must run on normal worker nodes.

Procedure

1. Go to Services page in IBM Fusion user interface.
2. In the Available section, click the Backup & Restore Agent tile.
3. In the Backup & Restore page, go through the features and capabilities of the service and click Install.
4. In the Install service window, select a Storage class that is used for the service.
   Select `ocs-storagecluster-ceph-rbd`for Backup & Restore service. The internal data catalog requires a minimum of 200 GB for ReadWriteAfter storage.
5. In the Install service window, select a Storage class that is used for the service.
   so select a storage class that supports this criteria.
6. Enter a connection snippet that is generated from the backup hub cluster.
   Important: When you install Spoke from the user interface, use the snippet. Use YAML option only when you do an automated deployment outside the IBM Fusion user interface.
   For more information about establishing connection between the Hub and Spoke, see Establishing connection between hub and spoke.
7. Click Install.
   A validation is done to check whether the connection is possible. If connection failed message appears, check the message and take corrective action. The installation starts and a notification appears on the Services page. You can see the progress of the installation in the Services > Installed section. After the installation completes successfully, you can see the status as normal and a Get started link.
   After you enable the Backup & Restore, you can view the service version and health status. From the ellipsis menu, you can download logs and view documentation. After you successfully collect the logs, a success notification gets displayed. The notification disappears automatically after some time.

   The Backup & Restore menu in the spoke cluster includes the following sub-menus:
   - Topology
   - Backed up applications

   In case of other failures, go through the downloaded logs to understand the cause of the failure and fix the issue. For more information about service issues in IBM Fusion, see Backup & Restore service installation and upgrade issues.
8. Validate the installation:
   Note: Ensure that the Backup & Restore IBM Fusion service is enabled before validation.

   Verify that the Backup & Restore IBM Fusion HCI service operators from the OpenShift Container Platform web console:
   a. Go to Installed operators from OpenShift Container Platform web console.
   b. Select the project as **`ibm-backup-restore`**.
   c. Verify that the following operators show the status as **`succeeded`**.
      - Red Hat Integration - AMQ Streams
      - IBM Fusion Backup & Restore Hub
      - IBM Fusion Backup & Restore Spoke
      - OADP Operator
   d. Alternatively, you can verify the status of operators by running the following oc command:
      Note: More operators appear in the command line output than in the web console.

      ```
      oc get csv -n ibm-backup-restore
      ```

      A sample result of the oc command is as follows:

      ```
      NAME                                DISPLAY                                       VERSION     REPLACES                PHASE
      amqstreams.v2.3.0-1                 Red Hat Integration - AMQ Streams             2.3.0-1   amqstreams.v2.3.0-0
      Succeeded
      guardian-dm-operator.v2.5.0         IBM Fusion Backup and Restore Data Mover      2.5.0
      Succeeded
      guardian-dp-operator.v2.5.0         IBM Fusion Backup and Restore Data Protection 2.5.0
      Succeeded
      guardian-mongo-operator.v2.5.0      IBM Fusion Backup and Restore Mongo           2.5.0
      ```

```
Succeeded
ibm-dataprotectionagent.v2.5.0    IBM Fusion Backup and Restore Agent            2.5.0
Succeeded
ibm-dataprotectionserver.v2.5.0   IBM Fusion Backup and Restore Server           2.5.0
Succeeded
oadp-operator.v1.1.2              OADP Operator                                  1.1.2     oadp-operator.v1.1.1
Succeeded
redis-operator.v2.5.0             IBM Fusion Backup and Restore Redis            2.5.0
Succeeded
```

.Ensure that the status shows **Succeeded**.

Verify the Backup & Restore pods from the OpenShift Container Platform console:

a. Go to Workloads > Pods.

b. Select the namespace, where you installed IBM Fusion Backup & Restore. In this case, select **ibm-backup-restore** namespace.
It lists all the pods. Ensure that all pods are running.

c. Verify whether the following pods are running successfully:

```
amq-streams-cluster-operator
applicationsvc
backup-location-deployment
backup-service
backuppolicy-deployment
guardian-dm-controller-manager
guardian-dp-operator-controller-manager
guardian-kafka-cluster-entity-operator
guardian-kafka-cluster-kafka-0
guardian-kafka-cluster-kafka-1
guardian-kafka-cluster-kafka-2
guardian-kafka-cluster-zookeeper-0
guardian-kafka-cluster-zookeeper-1
guardian-kafka-cluster-zookeeper-2
guardian-minio-0
guardian-mongo-operator-controller-manager
ibm-dataprotectionagent-controller-manager
ibm-dataprotectionserver-catalog-ibm-backup-restore
ibm-dataprotectionserver-controller-manager
ibm-backup-restoreagent-controller-manager
ibm-backup-restoreserver-controller-manager
job-manager
mongodb-0
mongodb-1
mongodb-2
mongodb-ab-0
openshift-adp-controller-manager
redis-master-0
redis-operator-controller-manager
redis-replicas-0
redis-replicas-1
redis-replicas-2
transaction-manager
velero
```

d. Alternatively, you can verify the installation by running the following oc command:

```
oc get pods -n ibm-backup-restore
```

A sample result of the oc command is as follows:

```
NAME                                                              READY   STATUS      R
amq-streams-cluster-operator-v2.3.0-1-7d6fb79d84-jdkfh            1/1     Running     0
applicationsvc-55c9b4d6c9-6hdv7                                   1/1     Running     0
b0f64f9161e0882f278dde2eaa1ea9677f4a230a29180fcf21fc665761hvvxz   0/1     Completed   0
backup-location-deployment-6b565b856c-j4vjc                       1/1     Running     0
backup-service-54bf9988f6-47bpv                                   1/1     Running     0
backuppolicy-deployment-b997cc9bf-dfwnh                           1/1     Running     0
bc6176f08ef686ccde24395724b77ea07a586a0bd1fa27ebfd5d704d0dxv9pl   0/1     Completed   0
e349f7c16f02ad6c0c31e41ba2fb1750d5154b58537224d239fe47508872cj5   0/1     Completed   0
f3dd0cbe8cb98614cf163fc5372733148236ea2bdeb5efc6a5d5afe4c085qlk   0/1     Completed   0
ff53c6d827e0fd610a4392c4f9411beb0c785572d2fca1bda57e208650bwv2m   0/1     Completed   0
ffacc5cdaa1e0aa4f3b3c0021f3c87931aa7422f8415303d95457febc8nmzk7   0/1     Completed   0
guardian-dm-controller-manager-64d57bf9ff-28dqj                   2/2     Running     0
guardian-dp-operator-controller-manager-6f6d55f6f7-fhndb          2/2     Running     0
guardian-kafka-cluster-entity-operator-b59d699f7-5qxt8            3/3     Running     0
guardian-kafka-cluster-kafka-0                                    1/1     Running     0
guardian-kafka-cluster-kafka-1                                    1/1     Running     0
guardian-kafka-cluster-kafka-2                                    1/1     Running     0
guardian-kafka-cluster-zookeeper-0                                1/1     Running     0
guardian-kafka-cluster-zookeeper-1                                1/1     Running     0
guardian-kafka-cluster-zookeeper-2                                1/1     Running     0
guardian-minio-0                                                  1/1     Running     0
guardian-mongo-operator-controller-manager-6f47776cb4-s6tkm       2/2     Running     0
ibm-dataprotectionagent-controller-manager-54d66f7975-lgdgh       2/2     Running     0
ibm-dataprotectionserver-catalog-ibm-backup-restore-k967t         1/1     Running     0
ibm-dataprotectionserver-controller-manager-749554d89f-q6gmx      2/2     Running     0
job-manager-859484bfc5-fzpt8                                      1/1     Running     0
mongodb-0                                                         2/2     Running     1
mongodb-1                                                         2/2     Running     1
mongodb-2                                                         2/2     Running     1
mongodb-ab-0                                                      1/1     Running     0
openshift-adp-controller-manager-59fb9f86f4-gdbhb                 1/1     Running     0
redis-master-0                                                    1/1     Running     0
redis-operator-controller-manager-647cf89ff7-wl8w2                2/2     Running     0
redis-replicas-0                                                  1/1     Running     0
redis-replicas-1                                                  1/1     Running     0
```

```
redis-replicas-2                                          1/1     Running     0
transaction-manager-5d9b59cf9f-hdzjm                      2/2     Running     0
velero-777d65c9b7-455fv                                   1/1     Running     0
```

Verify the Backup & Restore IBM Fusion service installation status from the OpenShift Container Platform web console:

a. Go to Installed operators from OpenShift Container Platform web console.
b. Select the namespace as **ibm-backup-restore**.
c. Select IBM Fusion Backup & Restore server.
d. Click the Data Protection server tab and select Data Protection server.
e. Select the YAML tab.
f. In the status section, makes sure the following:
- **HealthStatuses** Shows the health status of all components listed. It may take 5 minutes or more for all components to show up as healthy.
- Ensure the install status shows Complete and **progressPercentage** as 100.

g. Alternatively, you can verify the installation status by running the following oc command:

```
oc describe dataprotectionserver dataprotectionserver -n ibm-backup-restore
```

A sample result of the oc command is as follows:

```
Status:
  Health Statuses:
    Service Name:  applicationservice
    Status:        Healthy
    Service Name:  backuplocation
    Status:        Healthy
    Service Name:  backuppolicy
    Status:        Healthy
    Service Name:  backupservice
    Status:        Healthy
    Service Name:  jobmanager
    Status:        Healthy
    Service Name:  backupagent
    Status:        Healthy
    Service Name:  mongo
    Status:        Healthy
    Service Name:  redis
    Status:        Healthy
    Service Name:  kafka
    Status:        Healthy
  Install Status:
    Progress Percentage:  100
    Retry On Failure:     false
    Status:               Completed
  Installed Version:      2.5.0
  Upgrade In Progress:    false
  Upgrade Status:
    Retry On Failure:  false
```

h. Check the status section at the end of the CR and make sure the following:
- The Install Status section shows the status of the install along with the percentage complete.
- The install status shows the install status as Completed when the installation is successfully completed.
- The health status section lists and shows components health status as healthy.
- Note: Individual component health statuses may show **Unknown** or **Degraded** for up to five minutes, and show a healthy status when installation is complete.

What to do next

- You can set the configuration parameters in ConfigMap guardian-configmap to change defaults for IBM Fusion Backup & restore agent. For more information about the parameters, see Backup & restore configuration parameters.
- Go to the Backup spoke cluster user interface > Overview page and click Launch Backup Hub to open the Backup hub.
  In the Backup & restore > Overview page, you have quick links to generate YAML, connect locations, create backup policies, and protect your applications.
- You can now begin to protect your IBM Fusion applications.
  1. Add location to determine whether the network verification is needed. For the procedure to add a location, see Adding backup storage location.
  2. Creating backup policy.
  3. Assigning backup policy.

Establishing connection between hub and spoke

Generate YAML from the hub cluster to establish mutual authentication between the hub and spoke clusters.

Procedure

1. Log in to the user interface of the hub cluster.
2. Click Backup & Restore > Overview.
3. Use any of the following methods to generate the YAML.

Option	Description

Option	Description
Use the Fusion UI	a. In the step 1: Connect your clusters (optional) section, click Connect clusters. b. From the drop-down list, select Use the Fusion UI. The connect your cluster > Use the Fusion UI window appears. The snippet starts to generate. c. After the snippet is generated successfully, click Copy snippet. d. Enter the copied connection snippet when prompted during the spoke installation. For the step in Spoke installation, see Enter connection snippet.
Automate deployment	a. In the step 1: Connect your clusters (optional) section, click Connect clusters. b. From the drop-down list, select Automate deployment. The connect your cluster > Automate deployment window appears. c. Select the storage class to use on the spoke where you want to run the YAML. d. Click Generate. The YAML starts to generate. e. In the Deploy using YAML section, click copy icon in Deployment YAML or click Download YAML. Alternatively, in the Or deploy using oc command section, click Copy command. Important: If you use a custom namespace, then do not use the Copy command because the encoded command is not editable and is of no value with a custom namespace. f. Apply the downloaded YAML or the copied **oc** command on the Spoke cluster.

4. To validate the connection, do the following steps:
   a. Click Backup & Restore > Topology.
   In the Topology page, you can see the hub cluster. In the Cluster table, you can see a list of spokes connected.
   b. In the Topology page > Cluster table, check whether there is an entry for the spoke and the Connection status column is Connected for it. The Service status is Normal.

Uninstalling Backup & Restore

Steps to uninstall IBM Fusion Backup & Restore by using command-line or command prompt interface.

Before you begin

- Ensure you have the following installed:
  - Bash 5.0 or higher
  - OpenShift® Container Platform command line 4.14 or higher
  - `jq` 1.6 or higher version.

About this task

The uninstall-backup-restore.sh script has two optional arguments `-u` and `-d`. Use the `-u` argument to uninstall agent when the hub still exists.

Procedure

1. Log in to your cluster.
2. From your browser, go to this URL- https://github.com/IBM/storage-fusion/blob/master/backup-restore/uninstall/uninstall-backup-restore.sh.
3. Download uninstall-backup-restore.sh and run the script.

What to do next

Note: On the Topology page, it can take 10 minutes for the Connection Status of the spoke cluster to change from `Connected` to Unknown. Though in the Backedup applications page, you can see the previous application backups from the spoke cluster, no new backups are possible.
After uninstall of agent or server, you may want to delete the connection from the system. You can remove the connection for a spoke from the hub user interface, but the agent user interface does not have this capability. When you try to remove a connection from a hub, check your firewall. For more information, see Remove connections.

Getting started

To quick start the service, define the StorageClass and make sure the VolumeSnapshotClass matches the StorageClass provisioner. Next, connect your clusters and set up the backup locations. Create and assign backup policies to the relevant applications, and regularly back up for continuity. This section also introduces you to the protection of the Backup & Restore service itself from data corruption and loss.

- **Quick start Backup & Restore service**
  The Overview page provides quick access to essential Backup & Restore service tasks, which enables you to quickly set up and use the service.
- **Service protection**
  The IBM Fusion Backup & Restore service protection involves the backup of the Backup & Restore data management service to a S3 bucket.

Quick start Backup & Restore service

The Overview page provides quick access to essential Backup & Restore service tasks, which enables you to quickly set up and use the service.

Get started

From the Get started pane, you can quickly add locations, create policies, and assign a policy to your applications.

Step 1: Connect your cluster
Create connections to the clusters that you provide backup services for. If you are only backing up applications in this cluster, you can skip this step. Use any of the following methods to generate the YAML. For more information about the options, see Connect clusters.

Step 2: Connect backup locations
Connect S3 compatible object storage locations that stores the backups.

Step 3: Create backup policies
Define backup frequency, storage location, and how long they are retained.

Step 4: Protect your applications
Assign backup policies to your applications.

Step 5: Protect your Backup & Restore service
Protect the backup service by backing up the control plane to an S3 bucket. You can restore the backup service in another cluster in the event that this cluster fails.

Explore capabilities

Application resiliency
Backup applications so they can be recovered in the event of cluster loss, data loss, or corruption. See Backup & Restore overview.

Application mobility
Move applications between different clusters by using Hub & Spoke cluster configurations. Application backups taken in one cluster can be restored to a separate cluster. See Restoring an application.

Isolate your systems
Remote S3 backup location separates access between backups and applications. Your backups are not compromised even when a workload cluster is compromised. See Install Backup & Restore.

Orchestrate a backup or restore
Use a custom, application-specific recipe to consistently automate and control the required backup or restore sequence of an application. See Orchestrate a backup or restore.

Service protection

The IBM Fusion Backup & Restore service protection involves the backup of the Backup & Restore data management service to a S3 bucket.

If the cluster or the storage containing metadata becomes unavailable, you can recover the Backup & Restore metadata required to restore application backups.

You can configure service backups by specifying locations, setting schedules, and managing credentials. Additionally, you can restore a service configuration from a backup, connect to various backup locations (such as Azure, IBM Cloud, and Amazon Web Services), enter credentials, and initiate the restore process.

For more information about how to configure and restore, see Configuring service backups and Recovering service backups.

Backup & Restore of your workloads

The Backup & Restore service is designed to protect a wide range of workloads, including virtual machines, Kubernetes applications, AI workloads, and various IBM Cloud Paks. Additionally, it can also back up the Backup & Restore data management service to an S3 bucket.

Backup of application and VMs: The workload includes the following data:

- PVs of the deployed workloads
- Namespace and cluster scope resources for deployed workloads
- Application VMs that use filesystem based PVCs are backed up and restored.
  Important: To backup application VMs, ensure that the following prerequisites are met:
  - Bare Metal cluster is available
  - OpenShift Virtualization feature is installed from the Operator Hub of OpenShift® Container Platform console.

  Support is available for VMs based on Filesystem PVCs (PersistentVolume has the `volumeMode: Filesystem` defined). The Block mode volumes (`volumeMode: Block`) support is also available.

  Note: The Backup & Restore uses Velero hooks that are present on an application or resource. This means that if your application has hooks, then they are automatically run by Velero at the time of backup or restore.

For more information about OpenShift Container Platform applications, see https://docs.openshift.com/.

- **Backup & Restore of your applications and virtual machines**
  To protect your applications, create storage location, storage policy, and then enroll your applications for backup and restore.
- **Service protection**
  The IBM Fusion Backup & Restore service protection involves the backup of the Backup & Restore data management service to a S3 bucket.
- **Self-service Backup & Restore**
  You can protect your namespace application with IBM Fusion Backup & Restore even as an application user without cluster rights or a user without IBM Fusion administration rights. Application owners can create IBM Fusion Backup & Restore Custom Resource (CRs) within the application's namespace to self-manage their application's backup and restore needs without the involvement of a cluster or IBM Fusion Backup & Restore administrator.

Backup & Restore of your applications and virtual machines

To protect your applications, create storage location, storage policy, and then enroll your applications for backup and restore.

You create backup storage locations and polices as a prerequisite to backup. Assign the policies to applications such that backups run based on the defined policy schedule. Only one backup per policy or application combination allowed at a time. Redundant backups are canceled. IBM Fusion Backup & Restore supports block level incremental backups, which detect change blocks and processes only the changes since the last snapshot. It is also applicable for backups of OpenShift® Container Platform virtualization VMs using block volumes.

- **Backup storage locations**
  You can create backups by assigning a backup policy to an application. The policy specifies how frequently to backup the application, and where to store the backup. The location where the backup is stored is called a backup location and it is a prerequisite to create a policy.
- **Backup policies**
  A backup policy is a set of rules and schedules that define how and when data is backed up. It outlines the frequency of backups, the type of data to be backed up, the retention period, and the storage location. The Backup & Restore service you can create and manage backup policies to ensure your data is protected according to your specific needs.
- **Running on demand backups**
  Backup policies are applied to applications to ensure scheduled backups occur at the specified frequency. However, there are instances where you may want to run a backup outside of the predefined schedule. The Backup & Restore provides you the option to initiate backups on demand.
- **Jobs**
  A job in the Backup & Restore service represents the process of data backup and recovery. You can monitor jobs, manage jobs, and troubleshooting any issues that may arise during the backup and recovery process.
- **Restoring an application**
  You can restore backed-up data to the same project, a different project, or even a different cluster, while specifying the desired target storage classes.
- **Backup and restore as code**
  Make use of the backup and restore services through custom resources (CRs) and integrate these functionalities with the application code. It manages application backups by using a declarative state and maintains your backup posture for an application alongside the application in Git.
- **Backup & restore service from OpenShift Container Platform**
  You can monitor and manage Backup & restore service related operations from the OpenShift Container Platform console.

Backup storage locations

You can create backups by assigning a backup policy to an application. The policy specifies how frequently to backup the application, and where to store the backup. The location where the backup is stored is called a backup location and it is a prerequisite to create a policy.

Local snapshots store the backup on the local cluster, but it is a good practice to specify an external object store as a backup location so that you do not lose backups upon local storage failures. IBM Fusion uses a secure connection to transfer the backups to the object storage provider of your choice.

Supports multiple backup storage locations:

You can backup your Red Hat® OpenShift® applications to local and object storage target locations, such as IBM Cloud®, Microsoft Azure, Amazon Web Services, and any S3 Compliant object storage.

For S3 compliant object storage, ensure that the ACL (access control list) is enabled, and that read permission is granted. Also, ensure that S3 is not in the same cluster or namespace as IBM Fusion Backup & Restore service. You can plan to use cluster services like MinIO or NooBaa but ensure to host them on different clusters. If your backups are in the same place as your primary data, then the data is not protected.
Important: Backup & Restore does not support OpenStack Swift of ANY version.
On the Locations page of the IBM Fusion user interface, you can configure at least one location to store your backed-up data. It supports a variety of object storage types and includes details such as name, status, type, used capacity, associated policies, and applications. You can perform actions like viewing details, editing credentials, and removing locations using the ellipsis menu.

To manage the backup location by using CRs, see Backup and restore as code.

- **Creating a secret**
  You must create a secret for S3 targets that use a self signed certificate before you create a backup storage location.
- **Adding a backup storage location**
  You can add new location from the Backup & restore page of the IBM Fusion user interface.
- **Managing backup storage location details**
  You can view location details, update credentials (like access key and secret key for AWS, or account name and account key for Azure), access the YAML configuration, and remove the location record.

Creating a secret

You must create a secret for S3 targets that use a self signed certificate before you create a backup storage location.

Whenever you create a backup storage location for S3 compliant storage, you have the option to specify a certificate to authenticate the connection.

1. Extract the certificate from an S3 compliant service to a file. Use the `openssl` command to extract the certificate into the file `tls.crt`.
   Note: Ensure that the file name must be `tls.crt`.

   ```
   openssl s_client -connect <s3-service-name>-<s3-service-namespace>.apps.<fusion-hostname>.<domainname>:443 -showcerts \
    | sed -n '/BEGIN CERTIFICATE/,/END CERTIFICATE/p' > tls.crt
   ```

   For example, use the `openssl` command to extract the certificate from the `minio` service in the `minio-ns` namespace on a IBM Fusion cluster.

   ```
   echo | openssl s_client -connect minio-minio-ns.apps.myfusionhostname.mydomain:443 2>&1 | sed -ne '/-BEGIN
   CERTIFICATE-/,/-END CERTIFICATE-/p' > tls.crt
   ```

2. Run the **OC** command to create a generic secret in the IBM Fusion namespace using the `tls.crt` file.

   ```
   oc create secret generic <secret-name> --type=opaque --from-file=tls.crt -n <fusion-namespace>
   ```

For example, in the default IBM Fusion namespace `ibm-spectrum-fusion-ns`.

```
oc create secret generic minio-cert-secret --type=opaque --from-file=tls.crt -n ibm-spectrum-fusion-ns
```

Note: Make a note of this secret name. You need it to create a backup storage location.

Adding a backup storage location

You can add new location from the Backup & restore page of the IBM Fusion user interface.

Before you begin

If you plan to use a certificate for a S3 compliant backup storage location, then create a secret with the certificate as a prerequisite. For the procedure to create a secret, see Creating a secret.

About this task

Create the backup storage location in the specified sequence. High-level steps to dd a backup storage location:

1. Go to Backup & restore > Locations, and click Add location.
2. Enter Location name, type of object storage, and credentials for the storage.
3. If applicable, enter the certificate's secret name for S3 compliant storage locations.
4. Click Add.

Note: IBM does not support the creation of two backup storage locations that have both identical endpoint and bucket names.

Procedure

1. Log in to IBM Fusion user interface.
2. From the menu, click Backup & restore > Locations.
3. In the Locations page, click Add location.
   The Add backup location wizard page is displayed.
4. In the Add backup location, enter the Location name.
5. Select the type of object storage backup location. The different location types are Azure (Microsoft Object Storage), IBM Cloud (IBM Object Storage), AWS (Amazon Object Storage), MCG/NooBaa (Red Hat Object Storage), S3Compliant (Any Object Storage), and Storage Protect.
   For the procedure to add MCG/NooBaa location, see Using MCG or Noobaa as a backup storage location. For the procedure to add Storage Protect location, see Using IBM Storage Protect as a backup storage location.
   Note: S3 buckets must not enable expiration policies. For more information about this known issue, see S3 buckets must not enable expiration policies. Also, the bucket must not have an archive rule set.
6. Click Next.
7. In the Login credentials section, enter the following credentials to connect IBM Fusion to your backup location: Endpoint, Bucket, Access key, Secret key. If the location is Azure (Microsoft Object Storage), then enter the Account name and Account key instead of Access key and Secret key. If the location is Amazon AWS, then you must also enter the Region.
   Example for AWS endpoint:

   ```
   https://s3.us-west-1.amazonaws.com
   ```

8. In the Certificate settings (optional) section, enter the Secret name for the certificate.
   Note:
   - This setting is applicable only when you create an S3 compliant backup storage location type.
   - The endpoint URL must be an HTTPS protocol with a trusted connection. If the endpoint URL contains the HTTPS protocol, then you need to enter the name of the secret that contains the SSL certificate.
   - If you did not create a secret before you create the backup storage location, then you cannot complete further steps. Ensure that you cancel the operation and go back to the create secret step. For the procedure to create a secret, see Creating a secret.
   - If you plan to use a S3 location, check whether your permissions are valid with the cloud provider for that particular endpoint and bucket or equivalent. For certificates, run the following `openssl` command to check whether a Subject Alternative Name (SAN) exists:

     ```
     openssl x509 -in <filepath> -text
     ```

     The output displays a SAN field. This SAN field must match the endpoint host of the S3 bucket.
   - Do not use "wildcard + self-signed certificate" as it is a major security risk. If installed system-wide, then on exposure, all encrypted communication can be decrypted.

   Warning: For security reasons, HTTP connections to S3 backup storage locations are disabled by default. Enable HTTP access only in non-production environments. IBM strongly recommends avoiding this option whenever possible. To allow insecure connections, set the `allowInsecureS3` property to `true` in the `DataProtectionAgent` custom resource.
   For example:

   ```
   apiVersion: dataprotectionagent.idp.ibm.com/v1
   kind: DataProtectionAgent
   metadata:
     name: dpagent
     namespace: ibm-backup-restore
   spec:
     transactionManager:
       allowInsecureS3: 'true'
   ```

9. Click Add.
   A success message gets displayed after adding the location.

- **Using MCG or Noobaa as a backup storage location**
  Create an MCG or Noobaa S3 target backup storage location.
- **Using IBM Storage Protect as a backup storage location**
  Backup OpenShift Container Platform applications to an IBM Storage Protect server and optionally make a copy of the data on tape.

Using MCG or Noobaa as a backup storage location

Create an MCG or Noobaa S3 target backup storage location.

Procedure

1. Log in to Red Hat® OpenShift® Container Platform.
2. Go to Storage > Object Bucket Claims.
3. In the Create ObjectBucketClaim page, create an Object Bucket claim with the following details: capture credential details.

   ObjectBucketClaim Name
   : Enter the name of the Object Bucket Claim. If you do not enter a value, a generic name gets generated.

   StorageClass
   : Select `openshift-storage.noobaa.io` class. It defines the object storage service and the bucket provision.

   BucketClass
   : Select `noobaa-default-bucket-class`.

4. Click Create.
5. Note down the Object Bucket Claim credentials like Endpoint, Bucket Name, Access Key, Secret key.
6. Extract the certificate from an S3 compliant service to a file. Use the `openssl` command to extract the certificate into the file `tls.crt`.
   Note: Ensure that the file name must be `tls.crt`.

   ```
   openssl s_client -connect <s3-service-name>-<s3-service-namespace>.apps.<fusion-hostname>.<domainname>:443 -showcerts \ |
   sed -n '/BEGIN CERTIFICATE/,/END CERTIFICATE/p' > tls.crt
   ```

   For example, use the `openssl`command to extract the certificate from the `MCG/Noobaa` service.

   ```
   export s3_url=$(oc get routes.route.openshift.io -n openshift-storage s3 -o jsonpath='{.spec.host}'):443
   echo $s3_url
   openssl s_client -connect $s3_url -showcerts \ | sed -n '/BEGIN CERTIFICATE/,/END CERTIFICATE/p' > tls.crt
   ```

7. Run the **OC** command to create a generic secret in the IBM Fusion namespace by using the `tls.crt` file.

   ```
   oc create secret generic <secret-name> --type=opaque --from-file=tls.crt -n <fusion-namespace>
   ```

   For example, in the default IBM Fusion namespace `ibm-spectrum-fusion-ns`:

   ```
   oc create secret generic bsl-cert --type=opaque --from-file=tls.crt -n ibm-spectrum-fusion-ns
   ```

8. Log in to IBM Fusion user interface.
9. From the menu, click Backup & restore > Locations.
10. In the Locations page, click Add location.
    The Add a backup location wizard page is displayed.
11. Enter the Login credentials and Certificate settings to create a backup storage location.
    In the Secret Name for Certificate field, enter the secret name created in step 7.
12. After the backup storage location is created, go to the Locations page and check whether the status is Connected.
13. If you want to create a S3 of NooBaa type, do the following steps:
    You can add the location as a separate NooBaa type instead of the S3, but you cannot do it from the UI.

    It requires additional steps to create the credentials secret and BSL YAML.
    a. Create a secret using the `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY` from NooBaa.

    ```
    kind: Secret
    apiVersion: v1
    metadata:
      name: <noobaa bsl secret>
      namespace: <fusion namespace>
      labels:
        dp.isf.ibm.com/ownedBy: fbsl
        dp.isf.ibm.com/provider-name: isf-backup-restore
    data:
      access-key-id: <base64-encoded AWS_ACCESS_KEY_ID>
      secret-access-key: <base64-encoded AWS_SECRET_ACCESS_KEY>
    type: Opaque
    ```

    If a YAML is used, encode the values using base64.
    b. Copy the location of the S3 route to use as the endpoint for the BSL CR. Use noobaa CLI to get the list of buckets to choose from:

    ```
    noobaa bucket list
    ```

    c. Create the BSL CR with type `noobaa`:

    ```
    apiVersion: data-protection.isf.ibm.com/v1alpha1
    kind: BackupStorageLocation
    metadata:
      name: <noobaa bsl>
    ```

```
  namespace: <fusion namespace>
  labels:
    dp.isf.ibm.com/provider-name: isf-backup-restore
spec:
  credentialName: <noobaa bsl secret>
  params:
    bucket: <bucket name>
    endpoint: '<noobaa s3 route>'
  privateCertificateName: <optional noobaa certificate secret>
  provider: isf-backup-restore
  type: noobaa
```

Using IBM Storage Protect as a backup storage location

Backup OpenShift® Container Platform applications to an IBM Storage Protect server and optionally make a copy of the data on tape.

Before you begin

Ensure that you have IBM Storage Protect 8.1.18 server or higher.
Before you implement this solution, see IBM Storage Protect documentation to understand expected recovery time of data from tape based on the total amount of managed data on the tape.

About this task

If the IBM Storage Protect server already has an object agent defined, you must recreate the object agent to ensure that the public certificate is generated correctly. To recreate the object agent, you must do the following steps to remove the current object agent:

1. From the IBM Storage Protect server, issue the command **`delete server <object_agent_name>`**.
2. Uninstall the IBM Storage Protect object agent on the system by using the ibmspecified command from the **`delete server`** output.
3. Delete the object agent directory under the server instance directory on the machine that hosts the IBM Storage Protect installation.
4. Proceed with the steps in the procedure to define the IBM Storage Protect object agent.

Procedure

1. Configure the object agent service on the IBM Storage Protect server.
   a. Run the following command on IBM Storage Protect server:

      ```
      setopt objectagentsancertificate yes
      ```

      Note:
      - If you are using a self-signed certificate, then you need this step to allow the object agent to create a self-signed certificate with the Subject Alternate Name (SAN).
      - If you are using a CA-signed certificate, then this step is not required.

   b. Create the object agent service by using the **`DEFINE SERVER`** command.
      For details about the command, see IBM Storage Protect 8.1.18 documentation.
      Note: If you are using a self-signed certificate, then you must ibmspecify the HLAddress in dotted decimal format.
   c. Configure an object client as documented in IBM Storage Protect documentation.
   d. Create an object storage bucket. You can create a bucket either by using the S3 API or by using the following MinIO client command:
      `mc mb`
2. Configure a backup location from the IBM Fusion interface.
   a. Location name is any name of your choice.
   b. Location type is S3 Compliant.
   c. Endpoint is the URL using the dotted IP address of the object agent client, which was configured on the IBM Storage Protect server.
   d. Bucket is the bucket that was configured on the IBM Storage Protect server.
   e. Access key and Secret key are the keys that were generated when the object client was created on the IBM Storage Protect server.
   f. Secret name for certificate to create a secret to store the IBM Storage Protect public certificate as documented in the Adding a location or Adding a backup storage location.

   After the backup location is created, create a backup policy using the new backup location and assign one or more applications to the backup policy.

3. Copy data to and from tape:
   a. Optionally, copy IBM ibmspectrum Protect data to tape:

      ```
      PROTECT STGPOOL
              Type=Local
      ```

      You can schedule this command to run periodically.
   b. To recover data that is no longer available in the primary storage pool, use the **`REPAIR STGPOOL`** command to recover the data from tape to the primary storage pool. For more information about the command, see IBM Storage Protect documentation.
   c. Initiate the restore request from IBM Fusion.

Managing backup storage location details

You can view location details, update credentials (like access key and secret key for AWS, or account name and account key for Azure), access the YAML configuration, and remove the location record.

View location details

From the ellipsis overflow menu of the location record, click Details to open the <Location name> slide out pane.

- Expand the Policies and Applications to see a list of associated polices and applications. The number that is mentioned within parenthesis for Policies and Applications indicate the total number of associated polices and applications respectively.
- Click the edit icon to update Access key and Secret key. For Azure, the values are Account name and Account key.
- Click the launch YAML icon to view the YAML.

The details about the location vary based on the type. For example, Amazon AWS includes the following details: Type, Status, Used capacity, Endpoint, Bucket, Region, and associated Polices and Applications. The Region is only applicable for AWS locations.

Add location

Click Add Location to add a new location. For more information about the procedure, see Adding a backup storage location.

Edit location details

From the ellipsis overflow menu, click Edit to update the values of Access key and Secret key. For Azure, the values are Account name and Account key.

Alternatively, you can also select the location that you want to edit and click the edit icon in the side pane to modify the location details.
Note: You cannot update bucket details except credentials because two backup storage locations with different names can share the same credentials and endpoint.

Remove a location

From the ellipsis overflow menu of a location record, click Remove to delete the location record. Click Remove in the Confirm remove window. This action removes the location as an available backup location from IBM Fusion. However, you can still access this backup location through your storage service provider.

Backup policies

A backup policy is a set of rules and schedules that define how and when data is backed up. It outlines the frequency of backups, the type of data to be backed up, the retention period, and the storage location. The Backup & Restore service you can create and manage backup policies to ensure your data is protected according to your specific needs.

The Policies page lists all policies. You can search for policy records based on the backup location or other keywords, configure the table display, and perform actions like viewing, editing, or deleting policies from the ellipsis overflow menu.

- **Creating backup policy**
  You can add new backup policies from the Backup page of the user interface.
- **Managing backup policy**
  Click Manage policies to manage policies for an application and initiate backup now.

Creating backup policy

You can add new backup policies from the Backup page of the user interface.

Before you begin

As a prerequisite to create a backup policy with object storage, you must first add a backup storage location.

For more information about configuring the target backup storage Location, see Backup storage locations.

About this task

To manage the backup polices and backup location by using CRs, see Backup and restore as code.

High-level steps to create a policy:

1. In the Policies page, click Add policy to open the Create a backup policy panel.
2. Enter the policy name and select the desired frequency (Hourly, Daily, Weekly, Monthly, or Custom) with specific times and days.
3. Select between In place snapshot or Object storage for the backup location.
4. Enter the retention period.
5. Click Create policy.

Procedure

1. Log in to IBM Fusion user interface.
2. From the menu, click Backup & Restore > Policies.
3. Click Add policy.
   The Create a backup policy slide out pane displays on the screen.
4. In the Create a backup policy pane, enter the following details to create a new backup policy that meets your business needs:
   a. Enter the Policy name.

b. Enter the Frequency details:
   i. Enter whether the frequency is Hourly, Daily, Weekly, Monthly, or Custom:

      Hourly
      Choose a time.

      Daily
      Choose a time.

      Weekly
      Select a day for Schedule and Choose a time.

      Monthly
      Choose a day from calendar and time.

      Custom
      Choose a specific day and time.

   ii. If you select frequency intervals of 1 day or more, the Time window section is visible on the screen for you to select a start time, end time, and time zone.
   Note: Consider the following points when you enter details in the Time Window:
   - The minimum time window is 4 hours from start to end time. An error message gets displayed whenever the time window is less than 4 hours.
   - When you change the start time, the end time is automatically adjusted to reflect the new start time plus the current time window duration.
   - When you change the end time, the duration between the start and end time is adjusted automatically, while the start time remains unchanged.
   - If the start and end times are identical, the duration is considered a 24-hour time window.

c. Select either In place snapshot or Object storage from Backup Locations.

   In place snapshot
   For In place snapshot policies, the resource backup information is kept locally on the cluster in the backup and restore install namespace of the MinIO storage. The VolumeSnapshots and VolumeSnapshotContent objects are kept in the application namespace. The VolumeSnapshotContent objects that are not namespaced get stored at cluster scope. The restore job fails whenever you delete the VolumeSnapshots or VolumeSnapshotContent objects. At expiration time, the resource backup in MinIO, VolumeSnapshots, and VolumeSnapshotContents get deleted. The deletion of the used StorageClasses or VolumeSnapshotClass objects, which are used in backup and restore also causes the expiration to fail.
   Note: Capacity for in place snapshots is not available.

   Object storage
   Using the Object Storage option, you to choose from a list of available object storage. You can also use the Search option to filter the location records.

d. Enter the retention period. It is the duration in which the backup copies exist in the storage location. Choose a value for its unit that can be days, weeks, months, or years.
   For example, 30 Days.

5. Click Create policy.
   A message gets displayed informing you about the successful creation of a new backup policy along with the policy name.

Managing backup policy

Click Manage policies to manage policies for an application and initiate backup now.

Procedure

1. Go to Backup & Restore > Backedup applications page.
2. Select one or more applications from the list.

Option	Description
Single application	a. Select an application. b. From the ellipsis overflow menu, select Manage policies. A Manage policies page is displayed with a list of all policies in the local cluster.

Multiple applications	a. Select multiple applications.
	b. Click Manage policies.
	A Manage policies page is displayed with a list of all policies in the local cluster. The following notification message is displayed:
	`Mixed policies`
	`the application selected do not share the same set of policies.`

3. To remove all policies, do the following steps:
   a. In the Manage policies page, click the Remove all policies toggle button.
   b. If you set the Remove all policies toggle button to on, then the policies table is disabled.
   c. In the Remove all assigned policies confirmation window, click OK to proceed.
4. Click Backup now to initiate an immediate backup of one or more applications.
5. Select one or more policies and click Save.

Setting up backup policies

After you create a policy, you can assign them to applications so that application data is backed up in the specified frequency and stored in the defined storage, such as object storage or in place snapshot. You can set up backups for applications of the local cluster from the Applications page. To setup backup for applications of both hub and spoke clusters, go to Backed up applications page.

Set up policies from the Applications page

Before you begin

For backup and restore to work, you must create volume snapshot class.

About this task

The high-level steps to set policies for applications are as follows:

1. Go to the Applications page.
2. Select one or more applications for which you want to assign a backup policy.
3. Assign one or more policies for the selected applications, and click Assign. You can also initiate an immediate backup.

Procedure

1. Go to Applications from the menu.
   For applications that do not have backup policy set, the Backup status columns of these records show No policy. If a policy is assigned, the name of the policy is displayed in the Policies column. If more than one backup policy is assigned to the application, the number of policies is displayed.
2. Select the check box from the Name column or click ellipsis menu and select Assign backup policy in the Policies column to assign a backup policy to a single application with no policy assigned.
   Note: If there are no policies assigned before, select the policy provider either Backup & Restore or Backup & Restore (legacy).
   To assign a policy to multiple applications, select the checkboxes in the Name column for the applications to which you want to assign the policy, and click Assign policy. The Assign policies window gets displayed and click Continue.
   The Assign backup policy slide out pane gets displayed.
3. In the Assign backup policy slide out pane, the Backup & Restore is selected by default as a policy provider and you can assign one or more backup policies to the selected application.
   The details of all available backup policies are displayed, such as location, locationtype, frequency, retention, and time.
4. Select Run backup now check box.
5. Click Assign to assign a backup policy.
   In the Applications page, you can observe that the Backup status of the selected application shows as Not backed up for applications that have not got backed up yet. After the backup set up is done, the Backup status of the selected application changes to Completed with a green check mark. Also, the Last backup time shows the backup time.
   Now you can check the status of your job from the Jobs tab.

Set up policies from Backed up applications page

Before you begin

As a prerequisite, you must have created policies.

- Ensure one or more policies exist.
- If you want the application VMs to be backed up and restored, then you need to ensure that the following prerequisites are met:
  - Bare Metal cluster is available
  - OpenShift Virtualization feature is installed from the Operator Hub of OpenShift® Container Platform console.

  Note: Support is available VMs based on Filesystem PVCs (PersistentVolume has the `volumeMode: Filesystem` defined). From 2.8 release, the Block mode volumes (`volumeMode: Block`) support is available as a technical preview.
  Recipes for VMs are not supported.

Ensure you are aware of the Backup & Restore service related issues before you protect your applications. For more information about issues and how to resolve them, see Troubleshooting Backup & Restore service issues.

Procedure

1. Go to Backup & Restore > Backed up applications.
2. Click Protect apps.
   The Protect applications wizard page gets displayed.
3. In the Select applications page, select a Hub or Spoke cluster on which you want to backup your applications.
   After you select the cluster, the number of unprotected applications on the cluster gets displayed. The applications table populates only unprotected applications and does not display applications that are already protected.
4. Select your application record(s). You can also select the entire list all together.
5. Click Next.
   The Assign policies wizard page gets displayed. The table includes Name, Location, Type, Frequency, Time, Retention, and Backup now column headings. The Type indicates the supported providers, such as Microsoft Azure, IBM Cloud®, S3 compliant object storage, Amazon Web Services, and Storage protect.
6. Select policies to assign to the applications. You can select a single to all policies for assignment.
7. Optionally, you can turn on the Backup now toggle for a record to immediately start the backup.
8. Click Assign.
   After the validation is successful, a success notification message is displayed. You can go to the Backed up applications page and search the applications table for the newly selected applications.

Managing your application workloads

Applications contain resources, such as microservices, databases, and other stateful data distributed across projects. IBM Fusion provides ways to do application-centric backup, which provides data loss protection for your applications.

Note: If IBM Storage Scale is used for storage, it must have a minimum of 5G PVC size to perform backup and restore operations.

Note: The Applications page shows only local cluster applications or remote DR applications. Remote Backup & Restore Spoke applications do not show up on the Hub cluster. To view Spoke applications, go to Backup & Restore > Backed up applications > Protect apps view.

Application list

The Applications page lists all your OpenShift® applications with their backup details, such as Name, Used, Capacity, Backup status, Last backup on, Success rate, and Policies.

Table 1. Application list

Application list	The Applications page lists all your OpenShift applications with their backup details, such as Name, Used, Capacity, Backup status, Last backup on, and Policies. Note: In the Applications page, if you continue to see applications that got deleted from OpenShift, check whether associated backups are available. If you no longer need to restore those backups, delete the Application CR instance of that application from the CRD.
Actions	View details Use the Search field to filter and find your record from the list. You can use the settings icon to customize the headings. To view the application details, click the Name link of the record or click View details in the ellipsis menu. Assign backup policy To assign backup policy, see Assign backup policy. Backup now To start the backup of an application right away, click Backup now in the ellipsis menu of the application record. Restore Restore the application backup. For the steps to restore an application resource, see Restoring an application.

Assign backup policy

Select applications with no policy assigned:

1. Select one or more application(s) and click Assign backup policy. Alternatively, for an application, click Assign backup policy from the ellipsis menu.
   The Assign a backup policy slide out pane gets displayed.
2. Select one or more backup policies. You can select Run backup now to initiate a backup. If you do not see any policy that suits your requirement, create a new policy. For the procedure to, create a new policy, see Managing backup policies for application workloads.
3. Click Assign .

Select multiple applications having a mix of policies:

1. Select one or more application(s) and click Assign backup policy. Alternatively, for an application, click Assign backup policy from the ellipsis menu.
   The Assign a backup policy slide out pane gets displayed.
2. In the Policy provider section, select which service you want to use as a policy provider.
3. Select one or more backup policies. You can select Run backup now to initiate a backup. If you do not see any policy that suits your requirement, create a new policy. For the procedure to, create a new policy, see Managing backup policies for application workloads.
4. Click Assign .
5. In the Confirm assignment confirmation page, go through the information about the number of applications that cannot be assigned as they use policies that are provided by another backup service.
6. Click OK.

Managing the application details

The application details are classified into Overview, Storage, Backups, and Resources tabs.

You can click the Assign a policy to open the Assign a backup policy page. For more information about Assign a backup policy page, see Setting up backup policies.

In addition, you can also click Assign policy to open the Assign a backup policy page. For more information about Assign a backup policy page, see Setting up backup policies.

Overview tab

The Overview tab includes the following details:

Table 1. Overview tab details

Element	Description
Storage	The Storage section provides a summary of how this application is using storage resources. It represents overall capacity and usage of the PVCs in a line chart. The capacity usage of storage classes is represented as a pie chart. For example, 1200 GiB is the overall capacity and 300 GiB is the used capacity.
Events	The Events section includes critical and recent events that occurred on the application. Click View all to see all the events that are associated to this application.
Backups	The Backups section provides a summary of backup jobs. The percentage of successful backups, timestamp of the recent successful backup, and total number of available backups and backup policies. The storage capacity usage from a policy perspective is represented as a pie chart. For example, there are 3 policies and 600 GiB overall capacity. The "check_app" policy usage is 400 GiB, the "object_policy" usage is 150 GiB, and the "client26" sewage is 50 GiB.
Inventory	The Inventory section displays a clickable list of storage and resources. The number within parenthesis for Storage and Resources headings indicate the total number of available storage and resources respectively.

Storage tab

The Storage tab includes the following details:

Table 2. Storage tab details

Element	Description
Capacity usage by PVCs	Capacity usage by PVCs section displays the current application capacity usage by PVCs.
Capacity by storage class	Capacity by storage class section represents in a pie chart the total capacity of the storage class and the amount of used storage in GiB.
Storage classes	Storage classes section individually lists all the storage classes along with the total number of PVCs, total capacity of the PVC, and the total used capacity in GiB. The number that is specified within parentheses is the total number of storage classes. Note: A Persistent Volume (PV) storage is dynamically provisioned by using the storage classes. For more information about how to create storage classes, see Dynamic provisioning of Persistent Volume in filesystem.
Persistent Volume Claims	The Persistent Volume Claims section lists PVCs distributed across the associated storage classes along with their used and total capacities. Note: The PVC is a storage request and it uses the PV resources. It also does claim checks to these resources. The following details of the PVCs are available in the list: • The Name of the storage class. • The PVC status of the storage class. The values are `Bound` and `Unbound`. • The Used and Capacity indicates the used storage and the total capacity in GiB. • The Storage class and PV indicates the associated storage class and PV. Use the Search text box to filter the records in the list. You can filter based on PVC status and storage class parameters. You can further refine your search based on the different values of these parameters. For example, you can filter PVC status based on Bound, unbound, or All. You can also use the settings icon to choose the columns to display. Click Reset to default if you want the original system settings.

Backups tab

The Backups tab includes the following details:

Note: If a backup is in progress for the application, then a message is displayed with details about the percentage completion of the backup job.

Table 3. Backups tab details

Element	Description
Backups list	The Backups section lists all backup jobs. By default, the following details about a backup are available in the list: • The Time of backup. Click the Time link of a record to view details. You can sort records based on Time. The most recent • The Status of the backup. The various statuses are namely, Snapshot in progress, In progress, Completed, Failed, PartiallyFailed, Failed Validation. • The Policy used. • The Location where it is backed up. • The Size (GiB). Additionally, you can select following details available under Edit columns section by clicking the setting icon. • The Location. • The Duration. • The End time.
Search	Use the Search text box to filter the records in the list. You can also use the settings icon to customize the column display. Click Reset to default if you want the original system settings. For advanced search on the records, click the filter icon and select values for Policy, Status, or Location from the drop-down lists. After selecting the values, click Done. Click Clear filters to clear the selection made in these drop-down lists.
Usage section	The Usage section provides total number of available backups and used capacity.
Backup policies section	For individual associated policies, it provides Retention, Used Capacity, Last backup, and Location. You can use the ellipsis overflow menu to view details, edit, or remove a policy association from the application. If you want to assign policies from this section, click Assign. For the procedure to assign, see Setting up backup policies.
Actions from the ellipsis overflow menu of a backed up record	The ellipsis menu of your backed up record includes the following options: View details and Delete backup. View details The details slide out pane is displayed: • The Backup details section contains the time, status, policy, location, and name of the application. If the backup is available, then the status is Available.The Size and Expiration information is added to the Backup details section. Click </> (Launch YAML icon) to open and view the YAML. Click (Restore icon) to restore the backed up data. If the restore is in progress, then the Restore icon is disabled. • The Job details section contains the Job ID and Elapsed time. • The Inventory, Backup sequence, and Data transfer phases are displayed for monitoring. The backup's inline notification is displayed. If an error occurs in another of the statuses, then click View job details to know more information about the job. Delete backup Click Delete backup to delete the backup record.

Resources tab

The Resources tab lists all the included resources of the selected application in a table format with Name, Type, and Label headers.

Use the Search text box to filter the records in the list. For advanced search, click the filter icon and select a value from the Type drop down to list all resources that belong only to a selected type.

You can also use the settings icon to choose the columns to display. Click Reset to default if you want the original system settings.

Actions that you can perform on an application details page

Click Actions and select any of the following operations:

Table 4. Actions available in a application details page

Manage backup policies	Click Manage backup policies to select one or more backup policies for data protection. In the Assign policy slide out pane, click the Frequency and select run backup now to do an immediate backup. It overrides the frequency set in the policy and starts the on-demand backup job immediately. However, the scheduled backups continue to run as set in the associated policy. After you select, click Done to complete the operation. If the listed policies do not match your requirement, then create a new policy from the Backup Policies page. A message is displayed to inform the success of the policy assignment operation. Note: If you have not assigned a policy to this application, the Actions and Restore buttons are not visible. Instead Assign policy button is available.
Backup now	Click Backup now to backup the application data. In the Backup now slide out pane, verify the application, policy, backed up data location, and retention period. If there is more than one policy associated to the application, then select one or more policies and click Run <n> backup policies. Here, `n` indicates the number of policies that are selected to run backup. In the confirmation window, click Run backup policy to initiate the backup job.
Restore	Click Restore to restore the application backup. A Restore Application <application name> slide out pane gets displayed. For more information about how to work with the slide out pane, see Restoring an application.

Note: The Application CR remains as long as there are PolicyAssignments or Backups associated with the application. The system checks for both, and the Application CR is retained until neither exists.To completely remove the application, you must delete all existing Backups and PolicyAssignments, and then delete the namespace.

Running on demand backups

Backup policies are applied to applications to ensure scheduled backups occur at the specified frequency. However, there are instances where you may want to run a backup outside of the predefined schedule. The Backup & Restore provides you the option to initiate backups on demand.

Before you begin

- When you restore an application from Hub to Spoke or Spoke to Hub cluster, ensure that the same StorageClass exists in both Hub and Spoke.
- In Backup & Restore, one hub cluster and more than one spoke clusters may be connected to manage application backups in multi-clusters. For various reasons, the OpenShift® Container Platform cluster may have problems and become unusable. If the cluster recovers after the expiry of the client cert, you must clean and setup to rejoin the recovered cluster to the Hub or Spoke. For the procedure to rejoin, see Reconnecting OpenShift Container Platform cluster.

About this task

- You can backup one policy or more policies.
- The label `velero.io/exclude-from-backup=true` is not supported by the Backup & Restore service.

Procedure

1. Go to Backup & Restore > Backedup applications page.
2. Select one or more applications from the list.

   Note: Applications using remote Global Data Platform storage are not supported for Backup & Restore.

Option	Description
Single application	a. Click the Backup now option from the ellipsis overflow menu of an application record. b. Click Backup in the Backup now confirmation window to start the backup operation of the application immediately. The `Initiating backup Starting backup on [app name]` notification is displayed.
Multiple applications	a. Select multiple applications and then click the global Backup now. When different policies are assigned for multiple records, then the Backup now window is displayed. b. Select the backup policies that are used to backup the selected applications. c. Click Backup. The Backup now confirmation window appears with a notification about the availability of multiple policies. d. In the Backup now confirmation window, you can optionally select Exclude applications with multiple policies. e. Click Backup in the Backup now confirmation window to start the backup operation of the application immediately. The `Initiating backup on [n] applications` notification is displayed. In the Backedup applications page, you can observe that the Backup status of the selected applications are in progress.

Jobs

A job in the Backup & Restore service represents the process of data backup and recovery. You can monitor jobs, manage jobs, and troubleshooting any issues that may arise during the backup and recovery process.

View backup and restore jobs

The Jobs page of the IBM Fusion user interface the backup and restore jobs are segregated in their respective tab pages.

In the Backups tab page, you can monitor the statuses of all backup jobs, which can be categorized as failed, pending, in-progress, canceled, or completed. You can sort these jobs based on the Started field. The Backups table provides a comprehensive overview of each job, including its name, associated cluster, application, status, time, policies, and recipes used. By clicking on the job name, you can access detailed information about each backup job. Similarly, you can view the list of all restore jobs in the Restore tab page.

In the job details, you can manage and track your backup and restore jobs, cancel ongoing jobs, rerun jobs, adjust job resources, and collect logs. It also enables you to quickly access detailed information about any specific job to investigate further or troubleshoot any issues.

In the Summary section, you can view the start time and duration of the job. You can toggle between the List view or Log view.

Within the Summary section, the backup jobs follow the Inventory, Backup sequence, and Data transfer order. The restore jobs follows the Inventory, Data transfer, and Restore sequence order.

Inventory
: In the Workloads section, you can see the number of deployments, deploymentConfigs, stateful sets, and pods.
In the Storage section, you can see the number of PVCs attached.

In the Resources section, you can see the number of routes, secrets, services, and configMaps.

You can also download the resource list to your local.

Backup sequence
: You can see the status of pre-hook, backing up PVCs, post-hook, and backing up etcd resources.

Data transfer
: You can view the backup location, backup size, upload speed, upload status, and the upload progress percentage.
Note: You can view the data transfer details only for the jobs such as backups to S3 object and restore from S3 object.

From the Actions menu, you can click View YAML to view the CR's YAML in the OpenShift® console.

Cancel jobs

In the Jobs page, click Cancel jobs to cancel an ongoing job or a job in the queue. Alternatively, you can cancel jobs by using the Backup or Restore CR. Add the following line to the CR specification to cancel jobs that are in queue or running state.

```
spec:
  jobControl: cancel
```

If you cancel during backup, the volume snapshots get removed as a part of cleanup.

If you issue a cancel during restore, the cleanup happens only when you choose a new or an existing namespace.
Note: When you cancel a restore job to the original namespace, cleanup does not happen and can result in a corrupt application after the cancellation completes.

Scale jobs

You can scale out the number of transaction manager pods to process more jobs at the same time. For example, run the following command to increase the number of agent pods to two:

```
oc scale deployments/transaction-manager --replicas=2 -n ibm-backup-restore
```

To the scale out from the OpenShift Container Platform console, do the following steps:

1. Change the project to the Backup & Restore install namespace. The default is `ibm-backup-restore`.
2. Search and select the `transaction-manager` deployment.
3. Click the scale up or down to increase or decrease the scale. It takes about 30 seconds for a replica to start in optimal conditions. Do not click multiple times instead wait for each pod to get to ready status.

Re-run jobs

From the Actions menu, select Run job again to restart a previously completed or failed job.

Download logs

Download the log package for the current cluster and optionally any spoke clusters. Select the Collect service logs option in the Actions menu and choose your spoke clusters and click Create. The logs provide a high-level overview of the status of the job, including start and end times, success or failure, and any error messages. They are useful for quickly checking the overall status of a job and identifying any immediate issues.

Restoring an application

You can restore backed-up data to the same project, a different project, or even a different cluster, while specifying the desired target storage classes.

Before you begin

- When you restore an application from the Hub cluster to a Spoke cluster, manually check whether the same storageclass that is used to define the PVCs on the Hub exists on the Spoke cluster.
- By default, if PVCs with the same name exist in the namespace, they get restored with a new name (original name + "-n" suffix with n=1,2,3). If you want to skip the restore of existing PVCs, manually create a Restore CR from the command line and set an optional `spec` field `skipExistingPVC` to true.
- You must have OpenShift® APIs for Data Protection (OADP) version 1.3.1.
- For the Datamover operator to restore CephFS and Global Data Platform 5.2.0+ snapshots by using the ReadOnlyMany access mode, ensure that the value of backingSnapshot is `true` in the storageclass.
- Manually modify the restore when the original application already exists on the cluster and creates conflicts with each other.

About this task

- You can use the following restore options for your applications:
  - From S3 to the same cluster and project
  - From S3 to the same cluster and existing project
  - From S3 to the same cluster and new project
  - From S3 to a different cluster
  - From S3 to IBM Software Hub.
  - Restore from snapshot
- Backups that are taken using an In place snapshot can be restored only to the same project, and not to any new or a different project that exists. If you try to restore a project that is deleted from Red Hat® OpenShift, then the restore to the same project with In place snapshot does not include PVCs.
- After you remove an Agent or Spoke connection from the Hub cluster, it does not show up in the Topology page. Also, you can restore the available (not expired) successful application backups with an object storage location from that cluster to a different cluster.
- When you do a restore operation, the restored files, directories, and mount points can have different permissions, group, and owner from the source that was backed up; it is an expected behavior of Red Hat OpenShift design. For more information, see OpenShift documentation and https://access.redhat.com/solutions/7007252. You can modify this behavior by using security context constraints (SCCs), parameters such as fsGroup, or a custom script or command such as **chmod** or **chown**.
- IBM Cloud Pak for Data supports backup and restore on both x86 and Power.
- IBM Cloud Paks instance backups cannot be created when continuous backup is enabled on the MongoDB service.
- You can restore backup data to a different StorageClass provisioner on an alternate cluster, for example, restore data that was originally backed-up from a CephFS provisioned StorageClass to a Scale provisioned StorageClass.

Procedure

1. In the menu, click Applications.
2. In the Applications page, search and open your application that you want to restore.
3. In the application details page, go to the Backups tab and from the ellipsis overflow menu of the backup record, click Restore option.
   Alternatively, go to the Backed up applications page and select Restore option from the ellipsis overflow menu of an application record.
   A Restore Application <application name> slide out window gets displayed.
4. In the Restore Application <application name>, select a Cluster destination where you want the application to be restored.
5. Select a Project destination for the application data.

   Single namespace to the same or new cluster
   For a single namespace, you can choose from any of the following options:
   Note: For the single namespace to same or new cluster, the option Choose existing project is only available for same cluster restores.
   - If you select the Use same project option, then click Next and proceed.
   - If you select Choose existing project option, then select a Project from the list and click Next to proceed.
   - If you select Create a new project option, then enter the name of the project and click Next to proceed.

   Multi-namespace to the same or new cluster
   For multiple namespaces, you can choose from any of the following options:
   - If you select the Use same project option, then click Next and proceed.
   - If you select Create a new project option, then select Project suffix and click Next to proceed.
     The Project suffix is used to avoid name conflicts by associating it with a specific namespace. For example, if you select enter `123` for Project suffix, then in the preview you can see `<Project name>-one-123`. You can also choose different combinations of available suffixes in the Suffix preview.

   The details in the Summary section vary based on your selections.
6. If you have not selected the backups, then select the backups that you want to restore and click Next.
7. If you do not want to use the existing Target storage class in your cluster to restore the PVCs, then select Change target storage class and enter a storage class of the same type to change it.
   By default, restore occurs on a matching storage class name. If **`target storage class`** is specified, then restore happens on PVCs from this class. If the original storage class is non-existent or **`target storage class`** is not provided, the default storage class of the cluster is used.
   Example scenarios where there may be a need to change the Target storage class are as follows:
   - If there is a requirement for a filesystem PVC with RWX access mode that a particular storage class (such as RBD) does not support, consider using an alternative storage class that meets the requirement.
   - An application may require a storage class with specific attributes, such as a particular compression or encryption setting.

   The restore does not overwrite existing PVCs instead new PVCs are created with an annotated name. Modify your application deployment to use the new PVCs.
8. In the Missing etcd resources section, select Include missing etcd resources to restore non-PVC application resources that are included in this backup but do not exist in this Red Hat OpenShift project.
   Note: If you do not select the Missing etcd resources, the application-specific pods that need deployment resources or jobs may not start.
9. If you select Only restore a subset of selected PVCs, then select all PVCs or choose individual PVCs from the selected backup to restore.
10. Check the Summary and confirm the changes after restore.
11. Click Restore.
    A message appears informing that the restore of the application from its backup is in progress. The status of the backup record in the Backups tab is initially set to Restoring and then it changes to Completed.
12. Go to the application to confirm the success of the restore operation.

What to do next

Storage classes that are used by applications on the source cluster must exist or be created on the target cluster before you attempt to restore the applications that use those classes.

Backup and restore as code

Make use of the backup and restore services through custom resources (CRs) and integrate these functionalities with the application code. It manages application backups by using a declarative state and maintains your backup posture for an application alongside the application in Git.

You can use CRs to deploy backup storage locations, establish backup policies to each cluster, assign policies to applications, and perform backup and restore operations for applications and their resources.
Note: If you are not currently in the same namespace, then include `-n <namespace>` in all commands. If you are already in the IBM Fusion namespace (`ibm-spectrum-fusion-ns`), then `-n <namespace>` is not necessary.

Backup storage location

If you want to create a backup storage location, then define the type of the S3 endpoint, the name of the target S3 bucket, credentials to connect to the endpoint, and some optional parameters based on the type.

Create a backup storage location

```
oc create -f <storagelocation.yaml>
```

Sample storage location YAML file:

```
apiVersion: v1
data:
  access-key-id: AMIATNJ3JEMKR6GAUOLN
  secret-access-key: vgm9AJPztPkOygBFpBp2UzEErLBelTcp3JPdPn9c
kind: Secret
metadata:
  name: backup-storage-location-example-secret-0
  namespace: ibm-spectrum-fusion-ns
---
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: BackupStorageLocation
metadata:

  name: backup-storage-location-example
  namespace: ibm-spectrum-fusion-ns
spec:
  type: aws
  credentialName:  backup-storage-location-example-secret-0
  provider: isf-backup-restore
  params:
    region: us-west-2
    bucket: bucket-name
    endpoint: https://s3.us-west-2.amazonaws.com
```

In this sample, a secret and backup storage locations are created.

Note: The secret is created before the triple dashes, and the backup storage location is created after the triple dashes in the YAML file.
Run the following command to generate the cloud data:

```
printf "[default]\naws_access_key_id=minio\naws_secret_access_key=minio123\n" | base64 -w 0
```

Replace the value of `aws_access_key_id` and `aws_secret_access_key` with your value.

Modify backup storage location

```
oc edit fbsl backup-storage-location-example
```

You can modify the backup storage location details.

Modify secret

```
oc edit secret backup-storage-location-example-secret-0
```

You can modify the secret details.

Delete backup storage location

```
oc delete fbsl backup-storage-location-example
```

Note: You cannot delete a backup storage location if there are any existing backup policies or backups associated with it. For successful backups, they must surpass their retention period before IBM Fusion can automatically remove them. However, failed backups can be manually removed through the IBM Fusion web console.

Delete secret

```
oc delete secret backup-storage-location-example-secret-0
```

Get all backup storage locations

```
oc get fbsl
```

Backup policy

Create a backup policy

```
oc create -f <backuppolicy.yaml>
```

or

```
oc apply -f <backuppolicy.yaml>
```

Sample backup policy YAML files:

- Backup policy to take backups daily:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: BackupPolicy
metadata:
  name: daily-policy
  namespace: ibm-spectrum-fusion-ns
spec:
  provider: isf-backup-restore
  backupStorageLocation: backup-storage-location-example
  retention:
    number: 10
    unit: days
  schedule:
    cron: "30 10 * * *"
    timezone: America/Los_Angeles
```

- Backup policy to take backups weekly from Monday through Friday:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: BackupPolicy
metadata:
  name: backup-policy-weekdays
  namespace: ibm-spectrum-fusion-ns
spec:
  provider: isf-backup-restore
  backupStorageLocation: backup-storage-location-example
  retention:
    number: 4
    unit: weeks
  schedule:
    cron: "30 20 * * 1,2,3,4,5"
    timezone: America/Los_Angeles
```

Edit a backup policy

```
oc edit backuppolicy <backuppolicy name>
```

You cannot modify the policy name and CR name.

Delete a backup policy

```
oc delete backuppolicy <backuppolicy name>
```

Example:

```
oc delete backuppolicy daily-policy
```

If at least one reference exists a **backupObject** or **PolicyAssignment** Object, then you cannot delete the policy.

Get the basic information about a backup policy

```
oc get backuppolicy <backuppolicy name>
```

Get all the information about a backup policy

```
oc describe backuppolicy daily-policy
```

Alternatively, you can use the following sample OC command:

```
oc describe backuppolicy/daily-policy
```

Get all backup policies

```
oc get backuppolicies
```

Assign a policy to an application

BackupPolicyAssignment is a call to create backups by associating a backup policy with an application. You must assign a policy to an application to schedule backups.

```
oc create -f <policyassignment.yaml>
```

Note: The appCluster is the name of the cluster where this application resides. It is optional and only required for remote Spoke applications to designate which Spoke cluster to run the job against. If it is not provided, the assumption is that the jobs are for an application on the Hub cluster.
Sample assignment YAML file for daily backup:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: PolicyAssignment
metadata:
  name: backup-policy-assignment-example
  namespace: ibm-spectrum-fusion-ns
spec:
  application: application-sample
  backupPolicy: daily-policy
  appCluster: fusion-cluster
```

Run the following command to get the name:

```
oc get cluster
```

Example:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: PolicyAssignment
metadata:

  name: filebrowser-20220919-1834111-isf-auto-ibmcos-backuppolicy-20220919-183411
  namespace: ibm-spectrum-fusion-ns
spec:
  application: filebrowser-20220919-1834111
  backupPolicy: isf-auto-ibmcos-backuppolicy-20220919-183411
  appCluster: fusion-cluster
  runNow: true
```

The corresponding backup CRD file is as follows:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: BackupPolicy
metadata:

  name: isf-auto-ibmcos-backuppolicy-20220919-183411
  namespace: ibm-spectrum-fusion-ns
spec:
  backupStorageLocation: isf-ibm-location-20220919-183411
  provider: isf-backup-restore
  retention:
    number: 1
    unit: days
  schedule:
    cron: 51 18 1 * *
    timezone: America/Los_Angeles
```

Edit policy assignment

```
oc edit policyassignment <policyassignment-name>
```

Delete policy assignment

```
oc delete policyassignment <policyassignment-name>
```

Backup and restore

Create backup CR for on-demand backup

Here, one time application backup is taken by using an on-demand backup policy.

```
oc create -f <CR_Ondemand_backup.yaml>
```

Sample YAML file:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: Backup
metadata:

  name: auto-fb-spoke1-20230702-1122581-azure-dpcos1-daily-202307102004
  namespace: ibm-spectrum-fusion-ns
spec:
  appCluster: apps.bnr-sno-sunshine.fusion-sno-ibm.com
  application: auto-fb-spoke1-20230702-1122581
  backupPolicy: azure-dpcos1-daily
```

Note: The appCluster is the name of the cluster where this application resides. It is optional and only required for remote Spoke applications to designate which Spoke cluster to run the job against. If it is not provided, the assumption is that the jobs are for an application on the Hub cluster.
Run the following command to get the name of the cluster where this application resides and add it as a value for appCluster:

```
oc get cluster
```

Delete backup request CR

A `DeleteBackupRequest` can be used to delete a backup CR and all the related resources (backup data, snapshot, and so on) of a backup CR.

```
oc create -f <delete_CR_Name.yaml>
```

Sample YAML file:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: DeleteBackupRequest
metadata:
  name: backup-example-delete-request
  namespace: ibm-spectrum-fusion-ns
spec:
  backup: backup-example
```

Restore an application

```
oc create -f <backup-restore.yaml>
```

Sample of YAML restore file:

The following sample restores the `backup-wordpress` backup.

For cross cluster restore, the `targetCluster` is provided. If `targetCluster` is not available, then default to the same cluster where the backup got originated from:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: Restore
metadata:
  dp.isf.ibm.com/provider-name: isf-backup-restore
  name: restore-complete-wordpress-1689030046
  namespace: ibm-spectrum-fusion-ns
spec:
  backup: complete-wordpress-ibm-do-not-use-202307102253
  objectsToRestore:
    RESOURCES:
    - ALL
    v1/persistentvolumeclaim:
    - complete-wordpress/mysql-pv-claim
    - complete-wordpress/wp-pv-claim
  targetCluster: guard-vpc-dog-98b7318c91b01bd72490e80cc2328915-0000.us-south.containers.appdomain.cloud
```

Restore an application to alternate namespaces

You can restore application to alternate namespaces on same cluster or another cluster.

For example, application A comprising of three namespaces namely ns1, ns2, and ns3 are backed up, then you need to specify the `namespaceMapping` field in Restore CR to restore this application to alternate namespace.

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: Restore
metadata:
  name: <custom-restore-name>
  namespace: ibm-spectrum-fusion-ns
spec:
  backup: <your-backup-job-name>
  namespaceMapping:
      ns1: ns1-new
      ns2: ns2-new
      ns3: ns3-new
```

Namespace variables automatically point to the correct namespaces during backup and restore. And it would be useful in scenarios, where you want to refer current namespace like exec hooks, mainly in restore cases.

Note: If a namespace or namespace field is missing in the `namespaceMapping`, then it restores to the original namespace.

An example of application that is backed up with three variables in Application CR.

```
apiVersion: application.isf.ibm.com/v1alpha1
kind: Application
...
spec:
  includedNamespaces:
    - ns1
    - ns2
    - ns3
  variables:
    - name: PARENT_NAMESPACE
      value: ns1
    - name: CHILD1_NAMESPACE
      value: ns2
    - name: CHILD2_NAMESPACE
      value: ns3
```

There are two methods to access these variables:

- The first method is using `${PARENT_NAMESPACE}`. It is valid only in backup scenario, and becomes invalid during restore as still referring to `ns1`.
- The second method is using new construct `${APP.${PARENT_NAMESPACE}.name}`, which substitutes namespace variables correctly. This refers to `ns1` in case of backup and `ns1-new` when restoring to alternate namespace.
  In a restore scenario within the same namespace, it refers to `ns1` only. This new construct can be used while specifying hook `namespace` field and exec `commands`.

List backups

```
oc get backups.data-protection.isf.ibm.com
```

List restores

```
oc get restores.data-protection.isf.ibm.com
```

List policy assignments

```
oc get policyassignments.data-protection.isf.ibm.com
```

Self-service Backup & Restore

Create a `RecoverBackup` CR

To create a `RecoverBackup` CR, see Self-service Backup & Restore.

Service protection

Note: The service protection is supported only through the user interface and not from the command line.

Retrieve a backup logs and resources information

The following commands use jq to process the JSON output.

Get the Backup & Restore namespaces with the following commands using the former to retrieve the latter. Skip this step if you know the namespace used in the environment.
Note: You can run these commands only from the hub cluster, even when the Backup CR is created in a spoke cluster.

```
oc get spectrumfusion -A --no-headers | cut -d" " -f1

oc -n <fusion namespace> get fusionserviceinstance ibm-backup-restore-service-instance -o json | jq -rc
'[(.spec.parameters[]|"\n",.name,.value)]|@csv' | tr -d '"' | grep ",namespace," | cut -d"," -f3
```

Port forward to the backup service, then use curl or a web browser with either URL.

```
oc port-forward svc/backup-service 9443:9443 -n <backup and restore namespace> &

curl -k <logURL or resourcesURL> | jq -r '[.logs] | @sh'
```

Alternatively, the following steps may be used to run curl from the backup service pod:

1. Get the backup service's pod on the Backup & Restore namespace.

   ```
   oc get pod -n <backup and restore namespace> --field-selector=status.phase=Running --selector=app=backup-service | grep
   backup-service | head -n 1 | cut -d ' ' -f1
   ```

2. Run the curl command on the backup service pod.

   ```
   oc exec -it <backup-service-pod ID> -n <backup and restore namespace> -- curl -k <logURL or resourceURL> | jq -r '[.logs]
   | @sh'
   ```

Example:

```
: oc get spectrumfusion -A --no-headers | cut -d" " -f1
> ibm-spectrum-fusion-ns

: oc -n ibm-spectrum-fusion-ns get fusionserviceinstance ibm-backup-restore-service-instance -o json | jq -rc
'[(.spec.parameters[]|"\n",.name,.value)]|@csv' | tr -d '"' | grep ",namespace," | cut -d"," -f3
> ibm-backup-restore

: oc get pod -n ibm-backup-restore --field-selector=status.phase=Running --selector=app=backup-service | grep backup-service |
head -n 1 | cut -d ' ' -f1
> backup-service-cd597966b-btkgq

: oc exec -it backup-service-cd597966b-btkgq -n ibm-backup-restore -- curl -k https://localhost:9443/dataprotection/backup-
service/v1/logs/c2ed99be-6daa-4296-876e-6a027197ef0c/recipe | jq -r '[.logs] | @sh'

: oc exec -it backup-service-cd597966b-btkgq -n ibm-backup-restore -- curl -k https://localhost:9443/dataprotection/backup-
service/v1/logs/c2ed99be-6daa-4296-876e-6a027197ef0c/resources | jq -r '[.logs] | @sh'
```

Backup & restore service from OpenShift Container Platform

You can monitor and manage Backup & restore service related operations from the OpenShift® Container Platform console.

In the OpenShift Container Platform console, go to Storage > Backup & Restore. The Backup & Restore page includes the following tabs:

- Overview tab
  From this tab page, you can monitor storage consumption, backup storage activity, backup coverage, and backup and restore jobs.
- Policies tab
  From this tab page, you can do the following operations:
  - Create a backup policy
  - Edit an existing policy
  - Delete an existing policy
- Locations tab
  From this tab page, you can do the following operations:
  - Create a new backup storage location
  - Edit credentials of a backup location
  - Remove an existing backup location
- Applications tab
  From this tab page, you can do the following operations:
  - Assign backup policy to applications
  - Restore backups of applications
- Jobs tab
  From this tab page, you can do the following operations:
  - Delete backup jobs
  - Collect logs
  - View job summary

At any point in time, you can click Go to Fusion Backup & Restore to go to the Backup & restore page of the IBM Fusion user interface.

Prerequisites

- Enable Backup & restore service in IBM Fusion.
  For the steps to enable Backup & restore service from IBM Fusion user interface, see Install Backup & Restore.
- After IBM Fusion is installed, the following notification gets displayed in the OpenShift Container Platform console:

  ```
  Web console update is available
  ```

Click Refresh web console link to refresh and view the Backup & restore service.

- The user interface works only for the supported OpenShift Container Platform 4.16 and 4.17. Do not try to access that user interface on any older OpenShift Container Platform versions.

Overview tab

The Overview tab page is a dashboard view to monitor Backup & restore service details.

Section	Description
Status	The Status section includes Statuses of Backup location and Applications. The Backup locations are either Active or inactive. The Active refers to the connected location status and inactive refers to not connected. The Applications subsection includes the following details: Backed up applications Applications with assigned policies Applications with recent backup Applications that are backed up and the recent or latest backup was successful (Completed or PartiallyFailed). Backup in progress Applications with recent or latest ongoing backup.
Backup & Restore jobs summary	The Backup & Restore jobs summary includes the following details: Queued jobs It includes the number of in progress and number of pending jobs. Success rate Note: The Success rate and other fields are dependent on the frequency drop down. The Success rate subsection displays the percentage of success rate and the number of jobs completed out of the total number of jobs. It also gives a break up of the number of completed, failed, and canceled jobs. The All, Backup, and Restore options display the completed, canceled and failed jobs accordingly. The Backup job count is also represented in a graph bar chart.
Backup summary	The Backup coverage and Most recent backup are represented in a circular chart. It also shows a legend of the color representation. The Backup coverage includes the following details: Backed up applications Applications with assigned policies Not backed up applications Applications without assigned policies The Most recent backup includes the following details: Backup completed Applications that are backed up and its recent or latest backup was successful (Completed or PartiallyFailed) Backup failed Applications that are backed up and recent or backup was not successful.
Storage consumption	The Object storage section includes the amount of GiB used.
Backup & Restore activity	The Backup & Restore activity lists ongoing and recent events related to application, backup, backup policy, backup storage location, policy assignment, and restore.

Policies tab

A backup policy defines a set of rules to backup application data. The following table describes the different sections and actions you can perform in the Policies tab page.

Section	Description or action
Backup policies table	The Backup policies table lists the policies with Name, Status, Location, Backup scheduled, Retention period, Application. You can search a backup policy record by its name from the list.
Backup policy record actions	From the ellipsis overflow menu of a policy record, you can do the following operations: • Edit backup policy • Edit YAML • Delete backup policy

Section	Description or action
Create backup policy	1. Click Create backup policy. The Create BackupPolicy page is displayed. 2. Enter the following details of the policy: • Enter the Policy name of the policy. • In the Backup location section, select whether you want In place snapshot or Object storage. If you select Object storage, then select the Location from the drop down list. • Select the Backup schedule for the backup. In Schedule, select the frequency. In At, select the time. In Time zone, select your zone. • In the Retention period, select number for Schedule and Duration in `Days`, `Weeks`, `Months` or `Years`. It is the duration to retain the backup. 3. Click Create.

Locations tab

A backup location is a prerequisite to create a policy. You must configure at least one location to store your backed-up data. IBM Fusion supports a variety of object storage types. The following table describes the different sections and actions you can perform in the Backup locations tab page.

Element or section	Description or action
Backup locations table	The Backup locations table includes Name, Status, Location type, Policies, Used capacity, and Protected applications.
Actions performed on a backup location record	From the ellipsis overflow menu of a backup location record, you can do the following operations: • Edit credentials 1. Update Endpoint, Access key, or Secret key values. 2. Click Save. • Edit YAML opens in YAML format. You can make changes and click Save. • Remove backup location In the confirmation window, click Remove to proceed with deletion.
Create a backup location	If you plan to use a certificate for a S3 compliant backup storage location, then as a prerequisite create a secret with the certificate as a prerequisite. For the procedure to create a secret, see Creating a secret. Note: IBM does not support the creation of two backup storage locations that have both identical endpoint and bucket names. 1. Click Create backup location. 2. Enter the details about the backup location: • Enter the Name of the backup location. • Select the Location Type of the backup location. The different location types are IBM Cloud (IBM Object Storage), Azure (Microsoft Object Storage), AWS (Amazon Object Storage), Spectrum Protect, and S3Compliant (Any Object Storage). Enter the following credentials to connect IBM Fusion to your backup location: Endpoint, Path/Bucket, Access key, and Secret key. If the location is Amazon Web Services, then you must also enter the Region. If the location is Microsoft Azure (Microsoft Object Storage), then enter the Account name and Account key instead of Access key and Secret key. • In the Certificate setting (optional), enter the Secret name for certificate.

Applications tab

Applications contain resources, such as microservices, databases, and other stateful data distributed across projects. IBM Fusion provides ways to do application-centric backup, which provides data loss protection for your applications. The following table describes the different sections and actions you can perform in the Applications tab page.

Element or section	Description or action
Applications table	The Applications table includes Name, Backup status, Backup policy, Location, Used/Total capacity, Last back up on, and Success rate. Here, the Backup status indicates the outcome of the most recent backup performed for an application. The Success rate indicates the number of successful backups performed for the application. It also indicates the availability of backup copies.

Element or section	Description or action
Actions performed on an application record	From the ellipsis overflow menu of an application record, you can do the following operations: • Assign backup policy • Restore It opens a Restore window. 1. In the Select a project destination section, select from the following options: Use the same project the application is already using Select the project in the current cluster where the application gets restored to. Restoring to an existing cluster delete the contents of the project and replaces it with the restored application. Use existing project Create new project If you select Create a new project, then enter a new project. Note: If you choose Existing project or `Create new project`, you can only specify a single namespace.You cannot restore multiple namespaces to a single alternate namespace. 2. Select a backup from the available list of backups. 3. Optionally, select Include missing etc resource to restore any etc resources that are not present in the existing OpenShift project. 4. Select the PVCs. By default, all PVCs in the back are restored. If you want to restore only certain volumes, then you can select individual PVCs. 5. Click Restore.
Assign backup policy	If you plan to use a certificate for a S3 compliant backup storage location, then as a prerequisite create a secret with the certificate as a prerequisite. For the procedure to create a secret, see Creating a secret. Note: IBM does not support the creation of two backup storage locations that have both identical endpoint and bucket names. 1. Click Assign backup policy. Alternatively, you can also click Assign backup policy from the ellipsis overflow menu of an application record. You can search for backup policy and filter records. 2. Select one or more polices. 3. If you want to initiate a backup right away, set the Run backup now toggle button to on. 4. Click Assign. If you set the Run backup now toggle button is set to on, the Backup status of the application in the Applications page is initially New. It then changes to Queued, Inventory in progress, Snapshot inprogress, and Complete. You can also check the Backup & Restore Jobs list in the Jobs tab page for similar status.

Jobs tab

The Jobs page displays the details of all the backup and restore jobs. The following table describes the different sections and actions you can perform in the Jobs tab page.

Element or section	Description or action
Backup & Restore jobs table	The Backup & Restore jobs table includes Name, Operation, Status, Start time, Completion time (Duration), Application, and Policy. You can filter the records based on Status and Operation. The Operation values are Backup and Restore. You can select a field and enter keywords in the Search to filter records. You can also choose to view the list of jobs in last 24 hours, last 7 Days, or last 30 days.
Actions to perform on a jobs record	The ellipsis overflow menu of a job record are as follows: • Delete backup job If you click Delete backup job, a confirmation window appears. If you are sure to delete the job, click Delete. • Collect logs Click Collect logs to go to the Jobs page of the IBM Fusion user interface. In the Jobs page, click Collect logs. • Job summary Click Jobs summary to go to the Jobs page of the IBM Fusion user interface. Here, you can view details like Summary (Inventory and Backup sequence) and Details (Duration, Created, Size, Policy, Application. Backup ID, Backup location).

Service protection

The IBM Fusion Backup & Restore service protection involves the backup of the Backup & Restore data management service to a S3 bucket.

In case of cluster failure, use the IBM Fusion Service Protection to recover all namespaces from the failed cluster into a new cluster. However, it assumes that the service protection backup got created and all application backups exist before the cluster failure. The high-level steps to backup and restore the service:

- Create a new instance of OpenShift® Container Platform cluster or use an exiting healthy cluster.
- Install Backup & Restore service.
- The Service Protection Recover service uses the backup to restore all resources used by the Backup & Restore service such as backup policy, location, policy assignment, Backup CRs.
- Restore all other applications from Backup CRs.

- **Configuring service backups**
  Backup a service configuration from the service backup.

- **Recovering service backups**
  Restore a service configuration from the service backup.

Configuring service backups

Backup a service configuration from the service backup.

About this task

In the Configure Service backups, select the location type and enter the credentials to access the location. After the discovery of existing service backups, the Service protection page is displayed with a list of service backups. If no service backups are available, define a schedule with frequency and retention period, and initiate a backup.
In Backup & Restore, one hub cluster and more than one spoke clusters can be connected to manage application backups in multi-clusters. For various reasons, the OpenShift® Container Platform cluster may have problems and become unusable. If the cluster recovers after the expiry of the client cert, you must clean and setup to rejoin the recovered cluster to the Hub or Spoke. For the procedure to rejoin, see Reconnecting OpenShift Container Platform cluster.

Note: Do not create virtual machines or PVCs on the IBM Fusion namespace, as it causes restoration failures of service protection backups.

Procedure

1. Go to Backup & Restore > Service protection.
   The Service protection page is displayed. Here, Configure Service backups and Recover service from backup tiles are available.
2. Select Configure Service backups.
   The Connect backup location wizard page is displayed.
3. In the Location type tab, select the location type and click Next.
   The available location types are Microsoft Azure, IBM Cloud®, S3 compliant object storage, Amazon Web Services, and Storage protect.
   The Location credentials tab page is displayed.
4. In the Location type tab, enter the credentials to connect to your location. Enter the endpoint and bucket to use for the service protection backups. For SSL secured object storage locations, enter the Secret name for certificate.
   There are two limitations that are not enforced:
   - While the user interface does not allow the created BSL to be assigned to any other policy, it is still possible to do so by using the CR. However, do not do this action.
   - The same endpoint or bucket must not be used for service protection configuration for more than one cluster deployments at a time.

   Note: After you set the location or bucket details, you cannot edit it later.
5. Click Add.
   Note: The discovery of existing service backups may take a minute or so. After the discovery completes, the BSL changes the status. For example, Connected.
   The Service protection page is displayed with a Service backups table.
6. If no service backups are available, click Define schedule.
   The Define schedule window is displayed.
7. In the Define schedule, enter the following details:

   Frequency
   : The frequency of backup. It can be Hourly, daily, Weekly, Monthly, Custom. For Custom, you can choose a value in minutes. Select the Timezone.

   Retention period
   : How long does the backup reside before automatic deletion. Select the number and its unit. For example, 30 Days.

   Initiate Service backup now
   : You can use this toggle to imitate the backup immediately.

8. Click Save.
   The Service protection page is displayed. If you had initiated service backup, then you can monitor the progress of the backup in the Service backups table. The Service backup table has Time, Status, Location, and Size. The Backup summary section is updated with details of Available backup, Used capacity, Last successful backup, and Backup policy.
9. If you had not initiated any backup, then click Backup now in the Service backup table.
   The Backup summary section is updated with details of Available backup, Used capacity, Last successful backup, and Backup policy.

Recovering service backups

Restore a service configuration from the service backup.

Before you begin

- Ensure that the IBM Fusion Backup & Restore (Legacy) service is not installed. This recovery replaces all existing objects with the objects in the service backup.
- If service protection requires rebuilding the hub cluster, backup your application data to a cloud-based storage. Also, backup any defined local S3 storage on your cluster to a cloud storage. The recovery procedure in this topic restores only the metadata for backups, which got saved to the cloud storage.
  For example, you have a MinIO storage service defined and configured with a backup storage location that uses the local storage. The MinIO application must have its cloud backup policy to restore the service first, and then any applications backed up to the local S3 storage.

  You can use the local storage for sensitive applications, but ensure that you save the database to the cloud or another S3 location within the firewall. Restore the local storage location from a cloud backup before you attempt to restore other applications backed up to that storage service.

- The IBM Fusion namespace must remain the same between the service backup and service restore. When you recover a service backup to a new IBM Fusion deployment, make sure that the deployment uses the same namespace as backup.

Procedure

1. Go to Backup & Restore > Service protection.
   The Service protection page is displayed. Here, Configure Service backups and Recover service from backup tiles are available.
2. Select Recover service from backup.
   The Add a backup location wizard page is displayed.
3. In the Location type tab, select the location type and click Next.
   The available location types are Microsoft Azure, IBM Cloud®, S3 compliant object storage, Amazon Web Services, and Storage protect.
   The Location credentials tab page is displayed.
4. In the Location credentials tab, enter the credentials to connect to your location. For SSL secured object storage locations, enter the Secret name for certificate.
   Provide the same S3 information or credentials as provided in the original Service Protection backup configuration.
5. Click Connect.
   The discovery process looks for existing service backups.

Option	Description
No service backups available	If no service backups are available, then `No service backups` message is displayed. Click Define schedule to configure. Note: It may take a while for the discovery of existing service backups to complete.
Service backups are available	Initiate restore by using either of the following methods: • Click Restore service. • From the ellipsis overflow menu of a service backup, select Restore. • Click Backup to display the Backup details slide out pane, and then click the Restore.

   The Restore service page is displayed.
6. Select a backup and click Restore.
   Note: This restore action reverts all Backup & Restore objects to the selected backup.
   When the Restore is in progress, the user interface disables Backup & Restore functions. However, you can still create CRs outside of the user interface.
7. Do the following steps based on the scenario existing on your cluster:

Option	Description
Empty cluster with no Backup & Restore	No action required
Empty cluster but Backup & Restore installed	a. Type `Recover`. b. Click Initiate service restore. Note: The recover action restores all backup policy, location, policy assignment, CRs including the ones used by Backup & Restore from service backup. However, any available Backup & Restore backups on the restore cluster get deleted. If you need them, copy the backup CRs.

   The Service protection page is displayed. A `Service restore is in progress` notification message is displayed. You can monitor the progress of restore in the Summary section. When the restore begins, the service protection page gets replaced by a job summary page for restore.

What to do next

1. Application backups are available to restore before the backup storage locations are connected. Wait for the backup storage location to be connected before you attempt to restore.
2. Storage classes that are used by applications on the original cluster must exist or be created on the restored cluster before you attempt to restore the applications that use those classes.
3. Scheduled backups resume after the service restore is complete. However, these backups fail until the applications are restored from a backup. You can ignore these failed backups. After the application is restored from a backup, subsequent backups will be successful.
4. When you do a service protection restore, the Backed up applications page does not display the applications on the Spoke clusters that are not reconnected to the new Hub cluster. After a Spoke is manually reconnected, it may take a while for the backed up applications on that Spoke to appear on the page.

Self-service Backup & Restore

You can protect your namespace application with IBM Fusion Backup & Restore even as an application user without cluster rights or a user without IBM Fusion administration rights. Application owners can create IBM Fusion Backup & Restore Custom Resource (CRs) within the application's namespace to self-manage their application's backup and restore needs without the involvement of a cluster or IBM Fusion Backup & Restore administrator.

Important: This feature is only supported for applications by using Backup & Restore Custom Resources (CRs) and not through IBM Fusion user interface.
All self-service Backup & Restore CRs, including scheduled Backup CRs exist in the namespace of the application.

To create Backup & Restore CRs in the application's namespace, you need to be an application namespace administrator, an application user, or service account with granted permission to the namespace. Application namespace administrators can directly create Backup & Restore CRs within their application namespace.

Non-admin application users and service accounts need the namespace administrator to first grant IBM Fusion Backup & Restore permissions through a standard OpenShift RoleBindings. The RoleBindings grant permissions for the user or service account in the application's namespace with the specific Backup & Restore Role.

Each of the Backup & Restore CRs have an 'admin', 'crdview', 'edit', and 'view' Roles available for them. They are named in the following format:

`<CR Name>.data-protection.isf.ibm.com-v1alpha1-<Role>`

The following Backup & Restore CRs have Role available for them:

Role	CR
`backuppolicies.data-protection.isf.ibm.com`	Backup Policy
`backups.data-protection.isf.ibm.com`	Backup
`backupstoragelocations.data-protection.isf.ibm.com`	Backup Storage Location
`deletebackuprequets.data-protection.isf.ibm.com`	Delete Backup Request
`policyassignments.data-protection.isf.ibm.com`	Policy Assignment

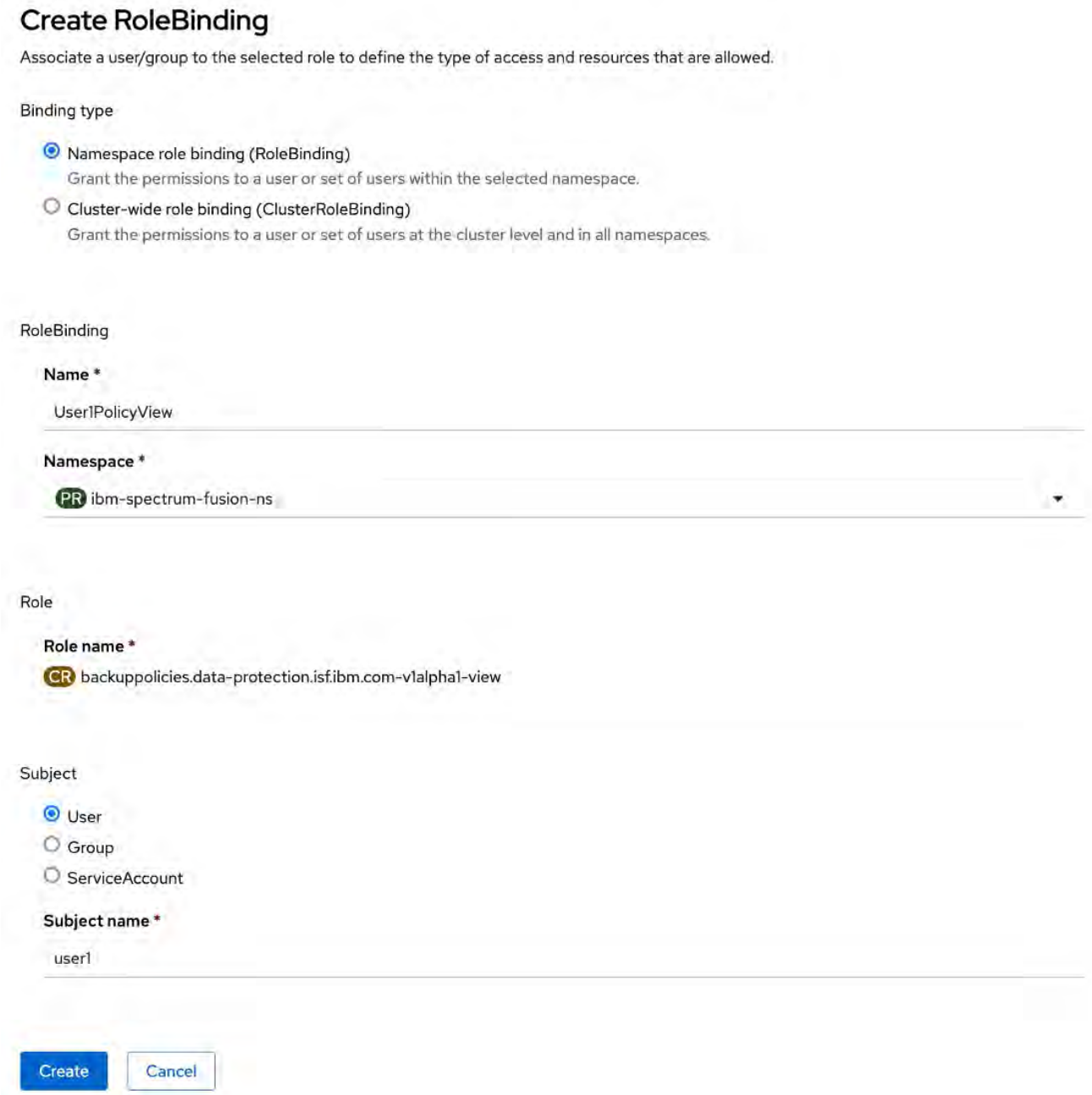
Role	CR
`recipes.spp-data-protection.isf.ibm.com`	Recipe
`recoverbackupcrs.data-protection.isf.ibm.com`	Recover Backup CRs
`restores.data-protection.isf.ibm.com`	Restore

For example, the IBM Fusion Backup CR has the following Roles:

- `backups.data-protection.isf.ibm.com-v1alpha1-admin`
- `backups.data-protection.isf.ibm.com-v1alpha1-crdview`
- `backups.data-protection.isf.ibm.com-v1alpha1-edit`
- `backups.data-protection.isf.ibm.com-v1alpha1-view`

If you want to reference `backuppolicy` or `backupstoragelocation` that is in another namespace, then specify it in "namespace/BP" or "namespace/BSL" format. The user must have permissions for the namespace that is specific for the BSL or BP.
RoleBindings can be created with a mix of permissions to achieve the desired level of management. For example, the application administrator wants to restrict Restore actions while allowing another user to perform all other Backup actions. To achieve this, the application administrator must create RoleBindings in the application's namespace for that other user with all Backup & Restore administrative roles except the Restore CR. To create the RoleBindings, see OpenShift documentation.

As an example, consider the following screen shot of the OpenShift® RoleBinding creation page showing the process of creating a RoleBinding for the user1 user identity in the namespace1 namespace with the `view` Backup & Restore PolicyAssignment Role.



The same RoleBinding can be created from the **oc** command line:

```
oc create rolebinding User1PolicyView --role=backuppolicies.data-protection.isf.ibm.com-v1alpha1-view --user=user1 -n
namespace1
```

Note: To retrieve logs and resources, see Backup and restore as code.

Service Accounts

It is a time-consuming process to create RoleBindings to multiple users individually. The Service Accounts provide another option for the self-service capability to a broad group of non-admin users. When creating a RoleBinding, you can select ServiceAccount option in the Subject section instead of an individual user. For more information about service accounts, see OpenShift documentation.

RecoverBackupCRs

If an application's namespace gets accidentally deleted or corrupted, the Backup & Restore CRs may prevent any potential recovery of the application from backups. It prevents the potential recovery of the application through an Backup CR.

Any successful **`RecoverBackupCRs`** action always recreates any prior Backup CRs. Recovery of other Backup & Restore CRs can be specified as options.

Example **`RecoverBackupCRs`** CR:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: RecoverBackupCRs
metadata:
   name: example
   namespace: <replacement namespace>
spec:
   application: <deleted application namespace>
   reassignPolicies: true
   recreateDeleteBackupRequestCRs: false
   recreateRestoreCRs: false
```

All spec fields are optional.

- **`application`**
  Name of the deleted application's namespace. If not specified, the Backup & Restore service makes use of the application name from the **`metadata.namespace`** field.
- **`reassignPolicies`**:
  Boolean to recreate any prior Policy Assignment CRs or not. Default is false.
- **`recreateDeleteBackupRequestCRs`**:
  Boolean to recreate any prior **`DeleteBackupRequest`** CRs or not. Default is false.
- **`recreateRestoreCRs`**
  Boolean to recreate any prior Restore CRs or not. Default is false.

To recover from an accidental deletion or corruption of the namespace or Backup & Restore CRs, do the following steps:

1. Create a replacement application namespace
2. Re-create the Backup & Restore RoleBindings that is same as the original namespace.
3. Create a **`RecoverBackupCRs`** CR in the replacement namespace that refers to the original namespace.

The Backup & Restore service detects the presence of this CR and performs verification checks to ensure that the administrative RoleBindings in the new namespace matches the deleted application. If it matches, then the Backup & Restore service recreates from its metadata all the relevant Backup & Restore. The desired Backup CR can then be used to restore the application from a prior backup.

Custom Backup & Restore Administrator

Another mode of operation for managing the Backup & Restore service is to grant a non-admin user the admin of the Backup & Restore service. To do this, create the RoleBindings of the user in the IBM Fusion namespace with the Backup & Restore admin Roles.

This user does not have IBM Fusion user interface access but can manage Backup & Restore activity of all applications on both hub and spoke clusters through Backup & Restore CRs within the IBM Fusion namespace.

Limitations

Limitations that are applicable for both Hub and Spoke:

- Only command line support is available
- Cross-cluster restore is not available
- Self-service jobs are not shown in the IBM Fusion UI
- Spoke self-service policies and backup storage locations are not shown in the IBM Fusion UI
- Applications with only self-service policies or completed self-service backup or restore jobs do not appear in the Backed up applications table.
- Applications configured with self-service policies do not appear in the Backed Up Applications > Protect Apps.
- On the Topology table, the backed up apps ratio and success rate may also be inaccurate as it includes self-service applications and self-service backup jobs.

Managing backup for multi-cluster environment

You can connect and manage cluster connections in IBM Fusion. It includes steps to establish connections and manage existing connections. You can also reconnect an OpenShift® Container Platform cluster either after a disaster recovery event or a temporary unavailability.

- **Connecting and managing cluster connections**
  You can connect clusters, manage actions on cluster records, and remove cluster configurations. You have options to connect clusters either through the Fusion UI or automated deployment.
- **Reconnecting OpenShift Container Platform cluster**
  The OpenShift Container Platform cluster that is connected in a disaster recovery relationship or a Backup & Restore service with Hub and Spoke can face disasters, leading to a temporary unusable state. After recovery, it can still display an `Unhealthy` or `UnManaged` status in remote clusters. In such instances, this cluster must be reconnected to its corresponding connections.
- **Disabling the connection**
  Disable the connection between Hub and Spoke.

Connecting and managing cluster connections

You can connect clusters, manage actions on cluster records, and remove cluster configurations. You have options to connect clusters either through the Fusion UI or automated deployment.

The Topology page displays the clusters' connections and you can perform the following activities:

- Connect clusters
- Actions on a cluster record
- Remove cluster configuration

Connect clusters

1. In the Backup & Restore menu, click Topology.
2. In the Topology page, click Connect clusters.
3. In the Connection method window, select Use the Fusion UI or Automate deployment.

Table 1. Cluster connection options

Option	Procedure
Use the Fusion UI	a. On your hub cluster, go to the ≥ Topology. b. Click Connect clusters. c. Select Use the Fusion UI. The connect your cluster > Use the Fusion UI window appears. The YAML starts to generate. d. After the YAML is generated successfully, click Copy snippet. e. Go to the Spoke cluster and install IBM Fusion. f. Go to the Services page and install Backup & Restore Agent. g. Paste the copied snippet to proceed with the agent install. Note: If you use a custom namespace, then do not use the **Copy** command because the encoded command is not editable and is of no value with a custom namespace.
Automate deployment	To connect multiple Spoke clusters: a. On your hub cluster, go to the ≥ Topology. b. Click Connect clusters. c. Select Automate deployment. The connect your cluster > Automate deployment window appears. d. Select the storage class to use on the spoke where you want to run the YAML. e. Click Generate. The YAML starts to generate. Here, you can either use the Deploy using YAML or Deploy using oc command option. f. In the Deploy using YAML option, copy or download the YAML. g. Apply the YAML on each of the Spoke clusters.

Actions on a cluster record

The Topology page lists the hub cluster and all spoke connections. Using the Actions, you can View details, View backed up apps, and remove connections.

In View details, a pane opens with the details of the cluster, namely Connection status, Service status, Job queue, any critical issues, and Backup success rate. You can also click View backed up apps in this pane to view applications that are local to a specific cluster.

In View backed up apps, view applications that are local to a specific cluster. The assumption here is that you assigned policies for these applications on the spoke.

You can also remove connections. See Remove connections.

Remove connections

You can search and remove connections.

From the Hub cluster, you can remove a Spoke or Agent cluster connection.
Note: The Backup & Restore service operations stops after the removal, however, all backups that are associated with this cluster remain available until they expire.
If the selected cluster is involved in a Regional DR, remove the Regional DR configuration before you remove the connection to the selected cluster.
If you remove the connection from the user interface, it removes the connection on both hub and spoke for that spoke. On the hub, there exists a connection for every spoke, so you must only remove the connection for the spoke to be uninstalled. There exists only one connection on the spoke or agent and hence you can remove it from the agent side safely. When you remove the connection on one cluster, the action removes it from both clusters.

Remove the connection first and then uninstall the agent.

1. Search for a connection.
2. From the ellipsis overflow menu of a connection record, click Remove connection.
   A Remove connection confirmation window is displayed.
3. Click Remove

A notification appears to confirm the successful removal of the connection. The cluster is not available in the table after the successful removal. From the removed Agent user interface, the following banner notification is displayed:

```
Unmanaged service The Backup & Restore Agent's connection to the Hub cluster has been removed.
```

Important: When a spoke cluster is removed, its application backups are deleted from storage when they expire. If the connection to the cluster is down but still configured, backups are retained until the connection is restored, and then deleted.
Alternatively, if you want to remove the connection from the hub side by using the **oc delete connection** command and more than one connection exists, then view the YAML and confirm the connected host that you want to remove.

If you want to remove the connection using **oc** commands, run the following commands:

```
oc get connection
```

```
NAME                     AGE
connection-7d9a76ce09    25h
```

```
oc delete connection connection-7d9a76ce09
```

```
connection.application.isf.ibm.com "connection-7d9a76ce09" deleted
```

Reconnecting OpenShift Container Platform cluster

The OpenShift® Container Platform cluster that is connected in a disaster recovery relationship or a Backup & Restore service with Hub and Spoke can face disasters, leading to a temporary unusable state. After recovery, it can still display an **Unhealthy** or **UnManaged** status in remote clusters. In such instances, this cluster must be reconnected to its corresponding connections.

About this task

If the cluster recovers before the expiration time, the cluster rejoins the connection automatically and no action is needed. However, if the cluster recovers after the expiration of the client cert, the connection must be cleaned and setup to rejoin the recovered cluster.

Procedure

1. Run the following example command to login to cluster-a and list the connection names.

   ```
   oc login to cluster-a
   ```

2. List the connection names.
   ```
   oc get connections
   ```
   It lists all connection CRs that are used for Backup & Restore and disaster recovery. The following example shows connections among clusters:
   - If cluster-a is in a DR relationship, a connection exists for the other cluster.
   - If cluster-a is a Backup & Restore hub, there exists a connection for each spoke.
   - If cluster-a is a Backup & Restore spoke, one connection to the hub exists.

   The output includes the API endpoint of cluster-b.
   Example output

   ```
   NAME                    API_ENDPOINT                                PHASE        HEALTHY_TO   HEALTHY_FROM   AGE
   connection-8414b9c84d   https://api.cluster-b.mydomain:6443   InProgress   Unhealthy    Unknown        25d
   %
   ```

3. Save the name of connection CR in cluster-a (a cluster in the hub and spoke or DR clusters) in **connection-<id>** format.
4. Clean the connection.
   For the procedure to clean the connection, see Disabling the connection.
5. Setup the connection between clusters.
   a. Log in to the cluster-b and get the bootstrap token for cluster-b.

      ```
      oc login to cluster-b
      ```

      ```
      oc create token isf-application-operator-cluster-bootstrap -n <Fusion Namespace of cluster-b>
      ```

      Note: The version of **oc** command line must be greater or equal to 4.11.
   b. Log in to cluster-a.

      ```
      oc login to cluster-a
      ```

   c. Create init secret on cluster-a with the bootstrap token and API endpoint of cluster-b.
      For example:

      ```
      apiVersion: v1
      kind: Secret
      metadata:
         name: <Init Secret Name>
         namespace: <Fusion Namespace of cluster-a>
      stringData:
         apiserver: <cluster-b API Endpoint>
         bootstrapToken: <cluster-b Token Generated in Step 3.a>
      ```

   d. Create connection CR on cluster-a with this init secret in spec:

      ```
      apiVersion: application.isf.ibm.com/v1
      kind: Connection
      metadata:
      ```

```
  name: <Connection Name Saved in Step 1>
  namespace: <Fusion Namespace of cluster-a>
spec:
  remoteCluster:
    apiEndpoint: <cluster-b API Endpoint>
    initSecretName: <Init Secret Name>
    connectionOperatorNamespace: <Fusion Namespace of cluster-b>
```

Disabling the connection

Disable the connection between Hub and Spoke.

About this task

In Hub and Spoke, there is a connection CR in each cluster that represents the connection to the another(remote) cluster. Delete connection CR in the Spoke or Hub. If the network is down, then use either method 1 or method 2 to manually delete the connection CR in another cluster also.

Important: Disabling a spoke connection results in the unassignment of policies from spoke applications, and no new backups are initiated subsequently. Custom Resources (CRs) of applications without backups are deleted, and these applications no longer appear in the user interface.

Method 1: Using command line or command prompt

Procedure

1. In the Spoke, use the following command to get all the connection CR:

   ```
   oc get connection -n <fusion-namespace> -o=custom-
   columns='NAME:.metadata.name,ClusterName:.metadata.annotations.connection\.isf\.ibm\.com/cluster-name'
   ```

   Example command:

   ```
   oc get connection -n ibm-spectrum-fusion-ns -o=custom-
   columns='NAME:.metadata.name,ClusterName:.metadata.annotations.connection\.isf\.ibm\.com/cluster-name'
   ```

   Example output:

   ```
   NAME                    ClusterName
   connection-44ae8529aa   cp4d-vpc-98b7318c91b01bd72490e80cc2328915-0000.us-east.containers.appdomain.cloud
   ```

2. Check the **ClusterName** in connection CR, and delete the connection CR which has the same ClusterName with Hub:

   ```
   oc delete connection <connection-name>  -n <fusion-namespace>
   ```

   Example command:

   ```
   oc delete connection connection-44ae8529aa -n ibm-spectrum-fusion-ns
   ```

Method 2: By using OpenShift Container Platform console

Procedure

1. Log in to OpenShift Container Platform console.
2. Go to Administration > CustomResourceDefinitions > connections.application.isf.ibm.com > Instances.
3. Get the connection instance whose cluster name in the annotation is the same as the hub cluster name.
4. Delete this connection instance.

Backup consistencies

- **Achieving data consistency for containers**
  IBM Fusion backs up application data consistently across all the Persistent Volumes (PVs). It ensures that the recovered data is error-free, reliable, and usable.
- **Achieving data consistency for virtual machine workloads**
  IBM Fusion backs up the data of the Red Hat® OpenShift® Virtualization virtual machines consistently.
- **Orchestrate a backup or restore**
  Use a custom, application-specific recipe to consistently automate and control the required backup or restore sequence of an application.

Achieving data consistency for containers

IBM Fusion backs up application data consistently across all the Persistent Volumes (PVs). It ensures that the recovered data is error-free, reliable, and usable.

The IBM Fusion supports application consistency and crash consistency approaches:

Application consistency
: In this approach, application I/O must be paused before a snapshot is taken by using recipes. Though it achieves the highest level of consistency, it comes at the cost of application downtime that can be disruptive. Depending on the application, the pause can be for a long time. To know more about how to create and use recipes, see Custom backup and restore workflows.

Crash consistency

If IBM Storage Scale is used for storage (and not Red Hat® OpenShift® Data Foundation), then you can consider crash consistency. In this approach, application I/O is not paused at all and hence results in zero disruption. It is achieved by what is called a consistency group. All the PVs of an application are placed in a consistency group. During backup, a snapshot is taken for all the PVs in the group at the same instant in time to ensure that all the PVs are consistent with each other. But the disadvantage of this approach is that the application might have some data in memory that is not flushed to the PVs and not included in the snapshot.
In IBM Fusion with IBM Storage Scale, all PVs in a namespace are placed in a consistency group automatically and you do not have to do anything.

Choose the type of data consistency that is appropriate for your application. If the application can restore itself without the data in memory with just what is there in the PVs, then backups are nondisruptive. Otherwise, you must pause I/O and disrupt the application.

Achieving data consistency for virtual machine workloads

IBM Fusion backs up the data of the Red Hat® OpenShift® Virtualization virtual machines consistently.

During virtual machine backups, consider the following options to provide data consistency within the virtual machine.

- **Crash consistency**- the backup includes data that is written to a disk. It is the equivalent to a virtual machine crash before a backup. To achieve this consistency, take a snapshot of the virtual machine PVCs without any additional orchestrations or hooks within the virtual machine.
- **File system consistency**- the file system freezes, writes, and flushes to the disk before a backup. The backup does not capture any data in the application buffers that is not yet written to the file system. Achieve this data consistency by using tools inside the virtual machine, which interacts with the file system to freeze write before a snapshot (backup) operation.
- **Application consistency**- data in memory buffers is written to disk before the application is backed up. It ensures that all application data is included in the backup. You can achieve this consistency by tools inside the virtual machine, which interacts with an application to take an application-specific action such as locking a database.

File system consistency

The following example diagram shows a virtual machine that runs within a namespace. Each virtual machine has a `virt-launcher` pod that runs the virtual machine instance. The `virt-launcher` pod runs an instance of `libvirt`, which is used to manage the virtual machine process.



When a namespace backup is requested, a virtual machine snapshot operation gets run. The `virt-launcher` pod includes Velero pre-snapshot (`pre.hook.backup`) and post-snapshot (`post.hoook.backup`) annotations to run the `kubevirt` tool `virt-freezer`. It communicates to a file system driver inside the Linux® or Microsoft® Windows virtual machine guest to freeze and unfreeze i/o operations around the virtual machine snapshot.

Configure and validate file system consistency for Linux virtual machines:

1. Ensure that `virt-launcher` pod has the appropriate Velero annotations:

```
pre.hook.backup.velero.io/command:
              '["/usr/bin/virt-freezer", "--freeze", "--name", "rhel9-lovely-louse",
              "--namespace", "test-vm-ns"]'

post.hook.backup.velero.io/command:
              '["/usr/bin/virt-freezer", "--unfreeze", "--name", "rhel9-lovely-louse",
              "--namespace", "test-vm-ns"]
```

2. Ensure that the QEMU guest agent (`qemu-guest-agent`) is installed in the Linux virtual machine. See OpenShift documentation.
3. Add the default service account security context constraint `kubevirt-controller`:

```
oc adm policy add-scc-to-user
              kubevirt-controller -z default -n <namespace>
```

4. Disable SELinux inside the virtual machine.

Configure and validate file system consistency for Microsoft® Windows virtual machines:

1. Ensure that `virt-launcher` pod has the appropriate Velero annotations:

```
pre.hook.backup.velero.io/command:
              '["/usr/bin/virt-freezer", "--freeze", "--name", "rhel9-lovely-louse",
              "--namespace", "test-vm-ns"]'

post.hook.backup.velero.io/command:
              '["/usr/bin/virt-freezer", "--unfreeze", "--name", "rhel9-lovely-louse",
              "--namespace", "test-vm-ns"]
```

2. Ensure that the QEMU guest agent (`qemu-ga-x86_64`) is installed in the Microsoft Windows virtual machine and that the “QEMU Guest Agent” network service is running. See OpenShift documentation.

3. Ensure that the `Volume Shadow Copy` service is running in the Microsoft Windows virtual machine.
4. Add the default service account security context `constraint` "`kubevirt-controller`":

```
oc adm policy
            add-scc-to-user kubevirt-controller -z default -n
        <namespace>
```

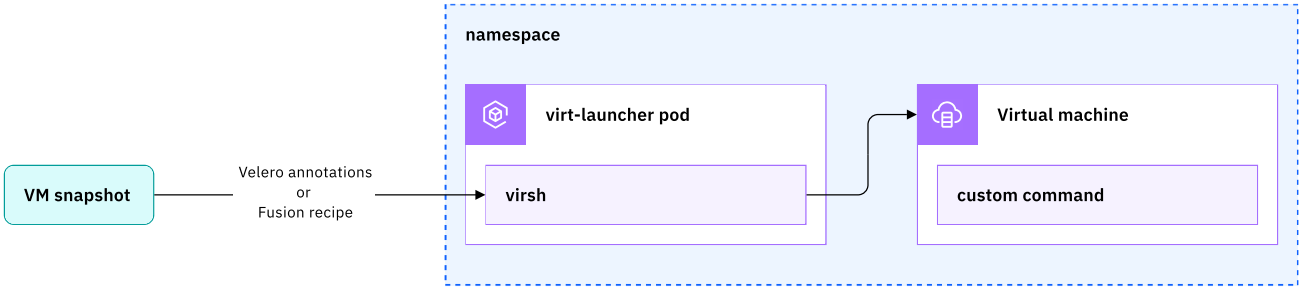
Crash consistency

You can achieve crash consistency by taking a snapshot of the virtual machine without engaging the QEMU guest agent to do a file system freeze. It an be done in one of following two ways:
Note: QEMU guest agent is a helper daemon. It is installed in the guest to exchange information between the host and guest, and to run command in the guest.

1. Remove the Velero pre-snapshot (`pre.hook.backup`) and post-snapshot (`post.hoook.backup`) annotations from the `virt-launcher` pod. This change does not persist across instantiations of the `virt-launcher` pod. The Velero annotations are automatically created along with a virtual machine and must be removed for each new instantiation of the `virt-launcher` pod.
2. Ensure that the QEMU guest agents are not installed or running in a virtual machine guest.

Application consistency

If you have a specific application consistency requirement that cannot be met by using the file system consistency, you can create a backup workflow that communicates directly with the application by using the virtual shell tool (`virsh`) instead of the `virt-freezer` tool. The following diagram depicts the scenario:



Consider the following alternative approaches run an application command by using `virsh`:

- Velero annotations to run an application command by using `virsh`:
  - You can directly modify the Velero annotations on the `virt-launcher` pod to run `virsh` (/usr/bin/virsh) instead of `virt-freezer` (/usr/bin/virt-freezer).
  - The modifications to the Velero annotations do not persist across instantiations of the `virt-launcher` pod. The Velero annotations are automatically created along with the virtual machine and must be modified for each new instantiation of the `virt-launcher` pod.
- Using IBM Fusion backup and restore recipes to run an application command by using `virsh`:
  Alternatively, you can create an IBM Fusion backup and restore recipe to run an application command by using `virsh`. Even though no differences exist in the outcome as compared to `virsh` by using the Velero annotations, consider the following points:
  - There might be instances wherein you must run more than one command to produce the wanted results. For example, issuing a command that locks a database for write and a second command to monitor or validate the lock. In these cases, a IBM Fusion recipe might be easier to work with.
  - If you decide to use an IBM Fusion recipe, the Velero annotations are also acceptable but it might produce undesirable results. For this reason, consider the removal of Velero annotations from the `virt-launcher` pod.
  - Use the recipe exec hooks to run `virsh` in the wanted sequence.

Orchestrate a backup or restore

Use a custom, application-specific recipe to consistently automate and control the required backup or restore sequence of an application.

For applications that support IBM Fusion Backup & Restore recipes, see Fusion Backup & Restore Recipes - Supported Applications.

Important: In Backup & Restore service, it is recommended to remove all Velero hooks from your applications before you use the recipe. Otherwise, the recipe and Velero hooks can contradict each other without your knowledge. If you want custom actions, then the recommendation is to use the recipe and not Velero hooks.

- **Custom backup and restore workflows**
  When you protect an application with Backup & Restore service, a default backup and restore workflow is used to protect an application. But while the backup and restore workflow is sufficient for some applications, there are some instances where you need to create a custom workflow for the backup and restore process.
- **Backup and restore of IBM Software Hub to the same or different cluster**
  IBM Software Hub is integrated with IBM Fusion , which enables you to create online backups and to restore them to the same cluster or to a different cluster.
- **IBM Fusion repository for recipes**
  The IBM Fusion public GitHub repository provides helpful utilities for IBM Fusion and allows members of the broader IBM Fusion community to share and contribute tools and knowledge.

Custom backup and restore workflows

When you protect an application with Backup & Restore service, a default backup and restore workflow is used to protect an application. But while the backup and restore workflow is sufficient for some applications, there are some instances where you need to create a custom workflow for the backup and restore process.

- The default backup workflow that is used by the Backup & Restore service is as follows:

1. Backup all namespace-scoped resources that are associated with the application and any dependent cluster-scoped resources.
2. Run any `Velero pre-backup` hooks that are specified through pod annotations in an arbitrary order.
3. Create snapshots of all PVCs associated with the application.
4. Run any `Velero post-backup` hooks that are specified through pod annotations in an arbitrary order.
5. If the backup policy specifies backup to an object storage, then copy the data to object storage.

- The default restore workflow that is used by the Backup & Restore service is as follows:
  1. Restore all PVCs.
  2. Restore all resources in an arbitrary order.
  3. Run `Velero post-restore` hooks that are specified through pod annotations in an arbitrary order.

You need a recipe to create a custom workflow for the backup and restore process. For more information, see Creating a Recipe.

- **Creating a Recipe**
  You need a recipe to create a custom workflow for the backup and restore process.
- **Assigning a Recipe**
  You can assign a recipe to your backup and restore operations.
- **Resource Transformation**
  Use the instructions to create a transform CR.

Creating a Recipe

You need a recipe to create a custom workflow for the backup and restore process.

The recipe custom resource has three basic specification elements:

1. Groups: A group defines a set of resources or PVCs that are processed together within a backup or restore. For example, a group can be a subset of specific resources that are specified with an exclude statement or an include statement.
2. Hooks: A hook can be used to start external scripts before and after snapshots, scale deployments up and down, or wait for certain conditions. For example, pods starting up and so on.
3. Workflows: A workflow defines the sequence of steps for a backup or restore operation, specifically the order in which the groups of resources and PVCs must be processed and any hooks that need to be applied.

For example:

```
apiVersion: spp-data-protection.isf.ibm.com/v1alpha1
kind: Recipe
metadata:
  name: wp-recipe
  namespace: wordpress
spec:
  appType: wordpress
  groups:
    - name: mysql_data
      type: volume
      labelSelector: app=wordpress
    - name: frontend
      labelSelector: tier in (frontend),app=wordpress
    - name: backend
      type: resource
      labelSelector: tier notin (frontend)
  hooks:
  - name: demoexechook
    type: exec
    namespace: ${GROUP.wp-app.namespace}
    nameSelector: wordpress-mysql.*
    ops:
    - name: pre
      command: >
        ["/bin/bash", "-c", "echo 'This is pretest' > /tmp/cfg_map_pre.txt"]
      container: mysql
    - name: post
      command: >
        ["/bin/bash", "-c", "echo 'This is posttest' > /tmp/cfg_map_post.txt"]
      container: mysql
  - name: demoscalehook
    type: scale
    namespace: ${GROUP.wp-app.namespace}
    selectResource: statefulset
    nameSelector: wordpress$
  workflows:
  - name: backup
    sequence:
    - hook: demoscalehook/down
    - group: frontend
    - group: backend
    - hook: demoscalehook/sync
    - hook: demoexechook/pre
    - group: mysql_data
    - hook: demoexechook/post
    - hook: demoscalehook/up
  - name: restore
    sequence:
    - group: mysql_data
    - group: backend
    - group: frontend
```

Specifying a group

Groups that are defined in the recipe refine the scope of the parent group, which is completely defined by the application CR. For more information about specifying groups in the application CR, see Assigning a Recipe CR with an application CR.
The following fields are mandatory for creating a group:

- **name**: name of the group that is specified in the workflow.
- **type**: type of group, either volume or resource.

The following fields are optional for creating a group:

- **LabelSelector**: Enables you to define multiple label selectors, and includes any resource or volume matching to the selectors in the backup and restore. For more on how to specify labels using or **LabelSelectors**, see Velero Backup API documentation .
- **backupRef**: Reference to a group used in a backup workflow. It is helpful for creating a subset of a backup group for a restore workflow.
- **includeResourceTypes**: list of resource type names to include.
  - It applies to resource groups only.
    For example, deployments.
  - If this field is not specified, then all resource types are included.
- **excludeResourceTypes**: list of resource type names to exclude.
  - It applies to resource groups only.
    For example, deployments.
  - **excludeResourceTypes** has precedence over **includeResourceTypes**.
- **essential**: determines whether the group represents volumes or resources, which are essential to ensuring that the backup can produce a successful restore (true or false).
  - Groups specified as essential (default) need to be processed successfully. Otherwise, the overall backup operation fails.
  - If the backup operation fails, a rollback is initiated, and the workflow is not processed any further.
  - For non-essential groups, an unsuccessful operation of the group is tolerated; the processing of the workflow continues, and the overall backup is reported as successful while only this operation is marked as failed.
  - The flag is only considered whether the workflow is configured with **fail-on=essential-error**. See the workflow section for details.
- **labelSelector**: select volumes and resources by label. For more information about on specifying labels, see Labels and Selectors.
- **nameSelector**: select volume or associated workloads by object name.
  - This field applies to volume groups only.
  - When combined with **labelSelector** or logic applies.
- **selectResource**: select which resource types the fields **labelSelector** and **nameSelector** must apply when you are selecting PVCs directly or indirectly through the workloads that the PVCs are associated with.
  - This field applies to volume groups only.
- **includeClusterResources**: determines how cluster-scoped resources are processed (true, false, or not specified).
  - **not specified**: include only cluster-scoped resources that are associated with the namespace-scoped resources that are included in the groups.
  - **false**: do not include any cluster-scoped resources.
  - **true**: include all cluster-scoped resources (still considering other clauses and label selectors).
  - This field applies to only resource groups.
- **includedNamespaces**: specify list of namespaces that are included.
  Note: Use **includedNamespaces** and **excludedNamespaces** when you have applications that can stretch multiple namespaces. If the application contains only one namespace, then the namespace that is used in the recipe is obtained from the application (applications.application.isf.ibm.com) resource. In this case, make sure that your recipe is more portable by avoiding including the namespace declaration explicitly.
- **excludedNamespaces**: specify a list of namespaces that are excluded.
  - This parameter has precedence over **includedNamespaces**.
    Note: Use **includedNamespaces** and **excludedNamespaces** when you have applications that can stretch multiple namespaces. If the application contains only one namespace, then the namespace that is used in the recipe is obtained from the application (applications.application.isf.ibm.com) resource. In this case, make sure that your recipe is more portable by avoiding including the namespace declaration explicitly.
- **restoreOverwriteResources**: Specify whether to overwrite resources during restore. By default, the value is false. If this field is set, then you must specify **backupRef**.
- **restoreStatus**: This field helps to preserves the statuses of the resources.
  For example:

```
- name: keycloak-resources
  type: resource
  includedNamespaces:
  - cp4i
  labelSelector: foundationservices.cloudpak.ibm.com=keycloak
  restoreStatus:
    includedResources:
    - integrationkeycloakclients.keycloak.integration.ibm.com
    - integrationkeycloakusers.keycloak.integration.ibm.com
```

  In this example, the statuses of the **integrationkeycloakclients.keycloak.integration.ibm.com** and **integrationkeycloakusers.keycloak.integration.ibm.com** are backed up and restored.

Specifying a hook

Hooks can be used to start external scripts before and after snapshots, scale deployments up and down, or wait for certain conditions (such as pods starting up).
The following types of hooks are supported:

1. **exec**: Start arbitrary commands and scripts within target containers.
2. **scale**: scale workload resources up and down.
   Note: New resource types are support for both **exec** and **scale** hooks:
   - For **exec** hook: **daemonset**
   - For **scale** hook: **daemonset**, **job** and **cronjob**.
     Both Singular and fully qualified resource types are supported for **exec** and **scale** hooks.

     For example:

Singular	Plural
pod	pods
deployment	apps/v1/deployments

Singular	Plural
satatefulset	apps/v1/satatefulsets
daemonset	apps/v1/daemonsets
job	batch/v1/jobs
cronjob	batch/v1/cronjobs

3. **check**: check for conditions on workload resources (such as **readiness**, **replicaCount** and so on).
   Important: You can have multiple logical expressions in check hooks using AND (&&) and OR (||) operators. It is also possible to create complex groupings using parentheses.
   For example:

   ```
   {$.spec.replicas} == {$.status.readyReplicas}
   ```

   ```
   {$.spec.replicas} == {$.status.readyReplicas} || {$.metadata.generation} == {$.status.observedGeneration} &&
   {$.spec.progressDeadlineSeconds} == {"500"}
   ```

   ```
   ({$.spec.replicas} == {$.status.readyReplicas}) || (({$.metadata.generation} == {$.status.observedGeneration}) &&
   {$.spec.progressDeadlineSeconds} == {"500"})
   ```

4. The following are the additional hooks that you can create in the IBM Fusion recipe improve its ability to perform application backup and restore operations. For more information, see Additional hooks.
   - Job hooks
   - Label hooks
   - Annotation hooks

The following fields are mandatory for creating a hook:

- **name**: the name of the hook that is specified in the workflow.
  - The name must be unique within the Recipe CR.
- **namespace**: the namespace to which the hook applies.
  - Namespace can be either specified directly (not recommended) or indirectly by the effective namespaces of a group **${GROUP..namespace}** or using a variable from the application CR **${VARIABLE_NAME**. In either case, you need to resolve to a single namespace.
- **type** : the type of hook, either exec, scale, or check.
- **selectResource**: workload resource type to that a hook applies. Specify the resource by using the fully qualified resource name. For example, **orchestrator.aiops.ibm.com/v1alpha1/installations**. The exception to this rule is that you can use the short names for commonly used resources pod, deployment, or **statefulset** (fully qualified resource name not required for these resources).
  - For exec hooks, defaults to pod.
  - This field is only required for hook types of scale or check.

  If you add a workload resource type that is specified by using the fully qualified resource name, add additional read permissions for the resource by updating the transaction manager **clusterroles** (**ibm-backup-restore** namespace) on the cluster (IBM Fusion hub or spoke) where the application is deployed. For example, if you plan to add the custom resource **orchestrator.aiops.ibm.com/v1alpha1/installations**, update the transaction manager **clusterroles** with the following command:

  ```
  oc edit clusterrole transaction-manager-ibm-backup-restore
  ```

  Add the following fields to the **clusterrole**:

  ```
  - verbs:
      - get
      - list
    apiGroups:
      - orchestrator.aiops.ibm.com
    resources:
      - installations
  ```

  Note: If you specify a resource of type pod, deployment, or **statefulset**, you do not have to update the transaction manager **clusterroles**.
- **labelSelector** or **orlabelSelector**: If specified, then the workload resource needs to match this label selector.
  Important:
  - The **labelSelector** includes **AND** condition for labels and both volume and resources of the recipe groups. It is comma separated (label) string that translates to AND operation later during resource selection.
  - The **orlabelSelector** includes **OR** condition for labels and both volume and resources of the recipe groups.
  - Either select **labelSelector** or **orlabelSelector**.
  - Either **labelSelector** or **nameSelector** is required. If both are specified, or logic applies.
- **nameSelector**: if specified, then the workload resource needs to match this expression.
  - Either **labelSelector** or **nameSelector** is required. If both are specified, or logic applies.

The following fields are optional for creating a hook:

- **singlePodOnly**: flag (true or false) that indicates whether to run a command on a single pod or on all pods that match the selector (when **selectResource=pod**). The hook is run on one of the matching pods only, but, which one is arbitrary.
  - For deployments and statefulsets (**selectResource=deployment or statefulset**), the option applies to each replicaset of the selected workload resources individually. For example, when two deployments are selected, the hook is run on one of the pods for each deployments replicaset, which pod is arbitrary.
  - This option applies only to exec hooks.
  - The default value is false.
- **essential**: flag (true or false) that indicates whether the hook is essential for a successful backup. Hooks that are defined as essential (which is the default) need to be processed successfully. Otherwise, the overall backup operation is considered as failed. In this case, a rollback is initiated, and the workflow is not processed any further.
  - For nonessential hooks, an unsuccessful execution is tolerated. The processing of the workflow continues, and the backup is reported as successful.
  - The flag is only considered whether the workflow is configured with **fail-on=essential-error**.
- **onError**: Determines the default behavior (fail or continue) if there is failures when an operation applies to multiple resources such as multiple pods. The option **fail** cancels the operation on the first occurrence of a failure, while **continue** continues processing of the remaining resources. In either case, the overall result of the operation is considered a failure.
- **timeout**: The default timeout (an integer value specified in seconds) applies to custom and built-in operations. If not specified in IBM Fusion 2.8.0, the default is 300 seconds. If not specified in IBM Fusion 2.8.1, the default is 10 days.

Ops specifications (optional)

- **ops**: set of operations that the hook can be started for.
  - Exec hooks provide you to specify any number of custom operations.
  - Other hook types have built-in operations.
- **ops/name**: name of the operation. This field needs to be unique within the hook.
- **ops/container**: the container where the command must be run.
  - The default is to pick an arbitrary container within the pod.
- **ops/command**: the command to run.
  - If you need multiple arguments, specify the command as a JSON array as single string, such as `["/usr/bin/uname", "-a"]`.
    Note: If you want to save embracing quotation marks so that you can use it within the command, use > to start a block scalar.
  - Variable substitution is applied, both intrinsic group namespace variables (`${GROUP...}`) and variables from application CR can be used.
  - For intrinsic group namespace variables, if there are multiple namespaces it yields a comma-separated list without spaces.
- **ops/onError**: specify fail or continue to provide the option to overwrite the identical option of the owning hook just for this operation. Defaults to the setting of the owning hook.
- **ops/timeout**: timeout (an integer value specified in seconds) applied to the specified operation. If specified, it overrides the timeout that is specified on the hook level. The hook is considered in error if the command exceeds the timeout.
  - Defaults to timeout set on hook level.
- **ops/inverseOp**: for rollback scenarios, it gives you the ability to rollback an operation by providing the name of another operation that reverses the effect of this operation.
  For example, resume a database would be the revert operation for a failure on a suspend a database operation.

Chks specifications (optional)

- **chks**: set of checks that the hook can apply to the target workload. Check that hooks provide an option to specify any number of custom checks based on conditions that are formulated as **JsonPath** like expressions.
- **chks/name**: name of the check. The name must be unique within the hook.
- **chks/condition**: the condition that needs to be true to release the hook. Conditions are specified as **JsonPath** like expressions.
  - The expression is expected to be a Boolean.
- **chks/onError**: specify fail or continue to overwrite the identical option of the owning hook just for this operation. Defaults to the setting of the owning hook.
- **chks/timeout**: timeout (an integer value specified in seconds) applied to the specified operation. If specified, it overrides the timeout that is specified on the hook level. The hook is considered in error if the command exceeds the timeout.
  Examples of specifying conditions as **JsonPath** expressions:
  1. Check to see if the number of available replicas matches the specified number of replicas.
     ```
     condition: "{$.spec.replicas} == {$.status.readyReplicas}"
     ```
  2. Check if a replica is ready.
     ```
     condition: "{$.spec.replicas} ==
     {$.status.readyReplicas}"

     condition: "{$.status.containerStatuses[0].ready}
     == {True}"
     ```

Specifying a workflow

A workflow defines the sequence of steps for both backup and restore operations. At least one backup workflow (named "backup") and one restore workflow (named "restore") must be specified in the recipe CR. More or alternative workflows can be specified that can be referenced in on-demand backup or restore requests (overriding the default ones "backup" and "restore").

The following fields are mandatory for creating a workflow:

- **name**: name of workflow.
  - The names "backup" and "restore" are reserved and implicitly used by default for backup or restore.
- **sequence**: sequence of steps to be performed.
  - Sequence can refer to a series of groups, resources, and hooks in a specific order. The actions are processed sequentially in the specified order.

The following fields are optional for creating a workflow:

- **sequence/step/group**: refer to a group specified.
- **sequence/step/hook**: refer to a hook and one of its operations.
- **failOn**: determines the behavior if failures when processing groups or hooks. The following is one of these values:
  - **any-error**: If this value is specified, the operation fails and performs defined rollback operations if any step of the workflow fails.
  - **essential-error**: if this value is specified, the operation fails and performs defined rollback operations if any step of the workflow that is defined as essential fails.
  - **full-error**: if this value is specified, the entire workflow is attempted. The workflow is only considered a complete failure if all essential steps fail.

- **Additional hooks**
  Use this section to understand the additional hooks and guidelines on how to create additional hooks.
- **Dynamic recipe**
  Use this section to create a dynamic recipe and guidelines on how to create it.

Additional hooks

Use this section to understand the additional hooks and guidelines on how to create additional hooks.

The following newly added additional hooks in the IBM Fusion recipe improve its ability to perform application backup and restore operations.

- Job hook
- Label hook
- Annotation hook

Job hook

A job hook supports to run custom operations during specific events, such as backups, restores or other workflows. It provides a flexible way to run long-running commands, provide dedicated environments for tasks, or perform batch processing and one-time tasks.

Note: Ensure that you use job string format either JSON or YAML.

Sample JobHook: JobSpec as YAML string:

```
hooks:
    - jobs:
        - forceCreate: true
          name: my-job1
          onError: fail
          timeout: 60
          inverseOp: ""
      name: job-hook-test1
      namespace: workload
      onError: fail
      timeout: 60
      type: job
  jobs:
    - my-job1: |
        apiVersion: batch/v1
        kind: Job
        metadata:
          name: sample-job
          labels:
            label-key1: value1
          annotations:
            annotation-key1: value2
        spec:
          template:
            spec:
              containers:
              - command:
                - perl
                - --version
                image: perl:5.34.0
                name: perl-version
              restartPolicy: Never
```

Sample JobHook: JobSpec as JSON string:

```
hooks:
    - jobs:
        - forceCreate: true
          name: my-job1
          onError: fail
          timeout: 60
          inverseOp: ""
      name: job-hook-test1
      namespace: workload
      onError: fail
      timeout: 60
      type: job
  jobs:
    - my-job1: |
            {
              "apiVersion": "batch/v1",
              "kind": "Job",
              "metadata": {
                "name": "sample-job",
                "labels": {
                  "label-key1": "value1"
                },
                "annotations": {
                  "annotation-key1": "value2"
                }
              },
              "spec": {
                "template": {
                  "spec": {
                    "containers": [
                      {
                        "command": [
                          "perl",
                          "--version"
                        ],
                        "image": "perl:5.34.0",
                        "name": "perl-version"
                      }
                    ],
                    "restartPolicy": "Never"
                  }
                }
              }
            }
```

The default fields for creating a job hook:

- Job hook jobs `forceCreate`: defaults to `false`
- Job hook jobs `timeout`: if not present, defaults to job hook timeout

- Job hook jobs `onError`: defaults to `fail`
- Job hook `timeout`: defaults to `864000` seconds
- Job string `name`: defaults to job hook job name
- Job string `namespace`: defaults to job hook namespace
- Job string `apiVersion`: defaults to `batch/v1`
- Job string `kind`: defaults to `Job`
- Job string missing `backoffLimit` value: defaults to `0`

Label hook

A label hook modifies a resource by applying a new label or removing an existing one.

Sample LabelHook:

```
hooks:
   - name: add-labels
     nameSelector: filebrowser*
     namespace: workload
     onError: fail
     timeout: 60
     ops:
       - labels: for-backup=true,myapp.io/app-name=filebrowser                # add labels
         name: filebrowser-resources-add-labels
         onError: fail
         overwrite: true
         timeout: 60
         inverseOp: filebrowser-resources-remove-labels
       - labels: for-backup-,myapp.io/app-name-                               # remove labels
         name: filebrowser-resources-remove-labels
         onError: fail
         overwrite: true
         timeout: 60
     selectResource: pod
     type: label
```

The default fields for creating a label hook:

- Ops `timeout`: defaults to label hook `timeout`
- Ops `onError`: defaults to `fail`
- Ops `overwrite`: defaults to `false`
- Label hook `timeout`: defaults to 864000 seconds
- Label hook `onError`: defaults to `fail`

Annotation hook

An annotation hook modifies a resource by applying a new annotation or removing an existing one.

Sample AnnotationHook:

```
hooks:
    - labelSelector: app=fb
      name: add-annotations
      namespace: workload
      onError: fail
      timeout: 60
      ops:
        - annotations: for-backup=true,myapp.io/app-name=filebrowser-workload         # add annotations
          name: filebrowser-resources-add-annotations
          onError: fail
          overwrite: true
          timeout: 60
          inverseOp: filebrowser-resources-remove-annotations
        - annotations: for-backup-,myapp.io/app-name-                                 # remove annotations
          name: filebrowser-resources-remove-annotations
          onError: fail
          overwrite: true
          timeout: 60
      selectResource: pod
      type: annotation
```

The default fields for creating an annotation hook:

- Ops `timeout`: defaults to annotation hook `timeout`
- Ops `onError`: defaults to `fail`
- Ops `overwrite`: defaults to `false`
- Annotation hook `timeout`: defaults to 864000 seconds
- Annotation hook `onError`: defaults to `fail`

Dynamic recipe

Use this section to create a dynamic recipe and guidelines on how to create it.

Dynamic recipe provides a capability to handle the backup and restore of applications with multiple sub-components dynamically. With a dynamic recipe, you can define a parent recipe that automatically discovers child recipes that provide extra workflow information, making your recipes more dynamic.

- Creating a parent recipe
- Creating a child recipe
- Adding new workflows with priorities
- Adding new workflows without a sequence
- Pre-recipe inventory workflow (Optional)
- Parallel backup hook execution
- Parallel restore hook execution

Some of the important considerations you should follow when you create a dynamic recipe:

- Ensure that child recipes correctly reference the parent recipe by using labels.
- You can define new workflows apart from reserved **backup** and **restore** workflows.
- If a child recipe uses the same workflow as the parent with a higher priority value, then the child recipe runs first.
- If a child recipe shares the same workflow and priority value as its parent, then the execution order is random but they run sequentially.
  The priority value can range from **-2,147,483,648 to 1,000,000,000**.

  For example:
  - A workflow with priority **-2,147,483,648** contain the lowest precedence.
  - A workflow with priority **1,000,000,000** contain the highest precedence.

Creating a parent recipe

In addition to the standard **backup** and **restore** **workflows**, you can define the new **workflows** in the dynamic recipe. These **workflows** enable dynamic behavior during the backup and restore processes.
Note: Parent and child can have same hook or group name but during recipe merge process child hook name is going to be prefixed automatically with child recipe name separated by period.
For example:

```
apiVersion: spp-data-protection.isf.ibm.com/v1alpha1
kind: Recipe
metadata:
  name: parent-recipe
  namespace: ibm-spectrum-fusion-ns
spec:
  ...
  hooks:
  ...
  workflows:
  - name: pre-hook
    priority: 4
    sequence:
    - hook: mysql-pod-exec/flush-tables-with-read-lock
  - name: post-hook
    priority: 4
    sequence:
    - hook: mysql-deployment-check/replicasReady
  - name: backup
    sequence:
    - group: mysql-resources
    - workflow: pre-hook
    - group: mysql-volumes
  - name: restore
    sequence:
    - group: mysql-volumes
    - group: mysql-resources
    - workflow: post-hook
```

Ensure that the names can be any valid string or similar to other workflow names like **backup** and **restore**. In this scenario, pre-hook is used in the backup workflow and post-hook is used in the **restore** workflow.
If a child recipe defines the same workflow, then it dynamically retrieves and runs.

Creating a child recipe

After a recipe linked to a another recipe through these labels **dp.isf.ibm.com/parent-recipe** and **dp.isf.ibm.com/parent-recipe-namespace**, then parent-child relationship gets created.

For example:

```
apiVersion: spp-data-protection.isf.ibm.com/v1alpha1
kind: Recipe
metadata:
  labels:
    dp.isf.ibm.com/parent-recipe: parent-recipe
    dp.isf.ibm.com/parent-recipe-namespace: ibm-spectrum-fusion-ns
  name: child-1
  namespace: ibm-spectrum-fusion-ns
spec:
  appType: mysql-child-1
  groups:
  ...
  hooks:
  ...
  workflows:
  - name: pre-hook
    priority: 5000
    sequence:
    - group: cm-child-1
    - hook: mysql-pod-exec-child-1/create-temp-file-pre
```

```
  - name: post-hook
    priority: 5000
    sequence:
    - hook: mysql-pod-exec-child-1/create-temp-file-post
    - group: cm-child-1
```

Adding new workflows with priorities

A child recipe can define workflows that are present in the parent recipe.

Adding a new workflow

For example, add a new workflow named `demo-workflow` with priority 1 in the parent recipe.

```
workflows:
  - name: demo-workflow
    priority: 1
    sequence:
      - hook: demo-hook/echo-test
```

Now, you can define the same workflow in the child recipe. The execution behavior depends on the assigned priority:

- High priority - Runs before the parent workflow
- Low priority - Runs after the parent workflow
- Same priority - Run order is arbitary (either the parent or child might run first).

Child workflow with higher priority

For example, the child recipe defines a demo-workflow with priority 5, then it runs before the parent workflow.

```
workflows:
  - name: demo-workflow
    priority: 5
    sequence:
      - hook: demo-child-hook/echo-child-test
```

Adding new workflows without a sequence

A parent recipe can define workflows that have no sequence. This allows child recipes to use these workflows and define their own sequences.

Defining a workflow in the parent recipe

For example, add a new workflow named `demo-workflow` in the parent recipe with no sequence and priority 0:

```
workflows:
  - name: demo-workflow
    priority: 0
    sequence: []
```

Defining a workflow in the child recipe

Now, the child recipe can define the same workflow and specify its own sequence:

```
workflows:
- name: demo-workflow
priority: 5
sequence:
- hook: demo-child-hook/echo-child-test1
- hook: demo-child-hook/echo-child-test2
```

Important:

- If the parent has only one child and parent workflow sequence is empty, then priority does not matter, and the child workflow executes as defined.
- If the parent has multiple children defining the same workflow, execution occurs according to the assigned priorities.

Pre-recipe inventory workflow (Optional)

This is an optional workflow for the parent recipe, run before the backup workflow.

- It allows an application to define and run its own custom exec hook in the parent recipe before any child recipe is merged into parent.
- Upon successful execution, IBM Fusion continues processing and takes the child recipe inventory for dynamic recipe execution.
- The main objective is to construct child recipes that reflect the current state of the application in the cluster.

For example:

```
apiVersion: spp-data-protection.isf.ibm.com/v1alpha1
kind: Recipe
metadata:
  name: parent-recipe
  namespace: ibm-spectrum-fusion-ns
spec:
  appType: mysql-ns
  groups:
  ...
  hooks:
  - name: my-child-recipes
    type: exec
    ...
    - name: "manage"
...
  workflows:
  - name: prerecipeinventory
    sequence:
```

```
    - hook: my-child-recipes/manage
  ...
```

Parallel backup hook execution

Running sequences in parallel within dynamic recipe workflows can significantly speed up the overall process. It is useful for workflows that involve tasks that depend heavily on I/O operations. If any one sequence fails, then the entire job aborts, but it waits for any currently running sequences in other parallel blocks to finish first.
Note: Undo operations are still performed sequentially, but in the reverse order of workflow execution.
Parallel hook execution can be requested in a IBM Fusion recipe by assigning the same **`priority`** to the workflow step in each recipe.

Key considerations

- Parallel execution is available for both backup and restore workflows.
- Parallel execution only can be used with dynamic recipes (when there is a parent recipe and at least one child recipe).
- If there are multiple recipes, consider running them simultaneously to increase efficiency.
  - Workflows with same priority run in parallel only if all sequences are hooks.
  - Workflows with same priority run sequentially if they have at least one **`group`** sequence within the workflow.
  - Workflows with different priorities run sequentially.
  - Workflows with mixed priorities: For example, there is a shared workflow that is defined in three recipes, two have the same priority and one is a different priority. In this particular case, the workflows with identical priorities run in parallel and the recipe with the different priority runs sequentially, before or after depending on the relative priorities.
- Number of parallel workers are controlled by flag **`hook-parallel-workers`** in **`guardian-configmap`** (ibm-backup-restore namespace). The default value is 1 and can be set to higher values with maximum of 35. Minimum value 0 have special meaning - internally translates to number of usable CPUs.

Parallel restore hook execution

IBM Fusion's parallel restore mechanism follows the same approach as its backup process. During restoration, the system dynamically merges saved parent-child recipes at runtime to build the final restore workflow. If a child recipe is not eligible for restoration to an alternate namespace, the system skips it during the merge. The status field in the Fusion Restore Custom Resource (CR) records details of any skipped recipes.

To perform a parallel restore with modified recovery steps (distinct from the saved parent recipes), create a custom **`Restore`** CR. Specify the new recipe **`name`** and **`namespace`** in the **`spec`** field.

Key considerations

Parallel restore adheres to all the key considerations outlined for backup. Additionally, it possesses specific capabilities that further contribute to successful application recovery.

- The system excludes child recipes that are labeled with **`dp.isf.ibm.com/alternate-namespace-restore=false`** when you target alternate namespaces.
- To introduce slight variations in restore steps, you can specify a new recipe **`name`** and **`namespace`** in the custom **`Restore`** CR. This existing capability is valuable especially during parallel restores.

Sample custom **`Restore`** CR

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: Restore
metadata:
  name: <custom-restore-name>
  namespace: <custom-restore-namespace>
spec:
  recipe:
    name: <new-parent-recipe-name>
    namespace: <new-parent-recipe-namespace>
```

Parallel backup and restore can significantly optimize data protection times when processing Fusion recipes, which have multiple activities that can be executed in parallel.

Assigning a Recipe

You can assign a recipe to your backup and restore operations.

1. Assign a recipe with a backup policy assignment CR for backup and restore operations as follows:

   Assign a recipe for backup operations

   You can assign a recipe to an application or applications by specify the recipe in the PolicyAssignment CR to assign it to one or more applications. You can specify a single recipe in a PolicyAssignment CR.
   To assign a recipe to an application, you can fix the PolicyAssignment CR by using the following example:

   ```
   oc -n ibm-spectrum-fusion-ns patch policyassignment POLICY-ASSIGNMENT-NAME --type merge -p '{"spec":{"recipe":
   {"name":"RECIPE_NAME", "namespace":"RECIPE_NAMESPACE", "apiVersion":"spp-data-protection.isf.ibm.com/v1alpha1"}}}'
   ```

   - **`POLICY_ASSIGNMENT-NAME`** is the name as specified in the PolicyAssignment CR.
   - **`RECIPE_NAME`** is the name of the recipe as specified in the Recipe CR.
   - **`RECIPE_NAMESPACE`** is the namespace where the Recipe CR is located.

   Assign a recipe for restore operations when restoring to the original namespace

   During a restore operation, a recipe must be started with the following precedence:
   - If there is a Recipe CR with the same name and in the same namespace as the Recipe CR that was used during the backup operation, it must take precedence. It can be helpful in situations when you want to specify a different workflow or different hooks during the restore.
   - If there is no Recipe CR with the same name and in the same namespace, the recipe that was used during the backup operation must be run. This information is stored as part of the backup metadata.

Assign a recipe for restore operations when restoring to a different namespace

If you are restoring to an alternative namespace, any namespaces that are specified in a hook cannot be run correctly as they refer to the original namespace. If you want the hooks to run in the alternative namespace, change the Recipe CR as mentioned in Assign a recipe for restore operations when restoring to the original namespace. So that any hook namespaces are referring to the alternative namespace.

2. Assign a recipe CR with an application CR as follows:

Reserved variables in application CR

The Backup & Restore service initializes the following reserved variables. Do not redefine these variables in your application CR, except for `ISF_ALLOWED_UNRESOLVED_VARS`.

- `ISF_BACKUP_UID`: the UUID of the backup job.
- `ISF_NAMESPACE_MAPPINGS`: namespace mappings in a comma-separated string format, accessible in exec hook commands.

  `"<source_namepace-1>:<target_namespace-1>,<source_namepace-2>:<target_namespace-2>,…"`

- `ISF_NOT_EQUAL_NAMESPACE_MAPPINGS`: an empty string when restoring to the same namespace, or the value of `ISF_NAMESPACE_MAPPINGS` when restoring to different namespaces.
- `SERVICE_NAMESPACE`: a predefined value, "`common-service`", used by cp4d.
- `${APP.${SERVICE_NAMESPACE}.nameEmptyIfNotMapped}`: checks if `SERVICE_NAMESPACE` exists, used by cp4d.
- `ISF_ALLOWED_UNRESOLVED_VARS`: This is Application variable used to skip verification for not found variables. Possible values are: `"true"` or `"false"`, default is `"false"`.

`ISF_ALLOWED_UNRESOLVED_VARS` is an application variable that skips verification for unresolved variables when set to `"true"`. Default is `"false"`. Set it to `"true"` if your application recipe uses shell variables and array syntax instead of substituting variables from the recipe.

```
spec:
  variables:
    - name: ISF_ALLOWED_UNRESOLVED_VARS
      value: 'true'
```

Working with application CRs for single namespace

For applications that only has a single namespace, there are no changes that are required in the application CR to support a recipe. The recipe is associated with the application, in this case with the PoliyAssignment CR, as defined in the Assign a recipe for backup operations.

Working with application CRs for applications that span multiple namespaces

If you have an application that spans multiple namespaces and want to use a recipe for the application, you need to specify all of the namespaces included in the application that is in the application CR.

- **Using variable substitutions**:
  - Variables can be used to permit a single recipe to be used in multiple, similar applications. Variables are defined in the application CR, and any number of variables can be defined.

    ```
    - variables:
       - name: <variable name>
           value: <variable value>
    ```

  - The variables that can be referenced within the Recipe CR.

    `${<variable name>}`

    Valid characters for variables are: a-zA-Z0-9_-

  - Each group exposes to the following variables automatically.
    - Effective namespaces
      These can be used within recipe by these patterns.

      ```
      ${GROUP.<groupname>.namespaces} - effective namespaces of group
      ${GROUP.<groupname>.namespace}  - effective namespace of group - single namespace enforced, (runtime) validation error otherwise
      ```

  - Variables can solely be used for the following fields:
    - Namespace field of hooks.
    - Within commands of exec hooks.

Tip: When creating a Recipe CR, place the Recipe CR in the `ibm-spectrum-fusion-ns` (or another common namespace). During a restore operation, you might need to change the recipe and don't want the recipe CR to be in the application namespace, as this namespace might not be available during restore.
For example,

```
apiVersion: application.isf.ibm.com/v1aplha1
kind: Application
metadata:
  name: wp-app1
  namespace: ibm-spectrum-fusion-ns
spec:
  appType: wordpress
  includedNamespaces:
    - wordpress
  variables:
  - name: WORDPRESS_NAMESPACE
    value: wordpress
---

apiVersion: spp-data-protection.isf.ibm.com/v1alpha1
kind: Recipe
metadata:
  name: wp-recipe
  namespace: wordpress
spec:
  appType: wordpress
  groups:
    - name: mysql_data
```

```
      type: volume
      labelSelector: app=wordpress
    - name: frontend
      type: resource
      labelSelector: tier in (frontend),app=wordpress
    - name: backend
      type: resource
      labelSelector: tier notin (frontend)
  hooks:
  - name: demoexechook
    type: exec
    namespace: ${WORDPRESS_NAMESPACE}
    nameSelector: wordpress-mysql.*
    ops:
    - name: pre
      command: >
        ["/bin/bash", "-c", "echo 'This is pretest' > /tmp/cfg_map_pre.txt"]
      container: mysql
    - name: post
      command: >
        ["/bin/bash", "-c", "echo 'This is posttest' > /tmp/cfg_map_post.txt"]
      container: mysql
  - name: demoscalehook
    type: scale
    namespace: ${WORDPRESS_NAMESPACE}
    selectResource: statefulset
    nameSelector: wordpress$
  workflows:
  - name: backup
    sequence:
    - hook: demoscalehook/down
    - group: frontend
    - group: backend
    - hook: demoexechook/pre
    - group: mysql_data
    - hook: demoexechook/post
    - hook: demoscalehook/up
  - name: restore
    sequence:
    - group: mysql_data
    - group: backend
    - group: frontend
```

Resource Transformation

Use the instructions to create a transform CR.

Before you begin

Ensure that the Backup & Restore service is installed. For steps to install Backup & Restore, Install Backup & Restore.

About this task

When restoring data to a different namespace or cluster, some resource definitions might need to be updated. For example, hardcoded storage classes or domain names that do not match the new environment.

While IBM Fusion Backup & Restore supports you to restore applications to alternative namespaces or storage classes. However, sometimes you need to change a resource before restoring it. This process is called a transformation.

Transformations are in-memory modifications that are applied to resources during the restore process, ensuring that they are correctly configured for the target environment before being written to disk or applied to the cluster.

Procedure

1. Define a transform CR that contains all the necessary transformations. The transform CR must be referenced in the restore CR, which ensures that the transformations get applied.
   - Create transform CR
   - Create restore CR with a transform CR
   - Apply the restore CR
2. A transform CR can consist of multiple transform rules. Each transform must have the following three sections:

   Name
   : The name of each transform rule must be unique.

   Patch type
   : There are four types of patches that can be defined:
     - `Json patch`
     - `Merge patch`
     - `Strategic patch`
     - `String patch`

   Subject
   : The subject specifies the resources on which the patch gets to apply.

- **groupResource** - Fully qualified resource name, such as
  - Core API group (v1) - pods, services, configmaps, secrets, persistentvolumeclaims, and so on.
  - Named API groups (apps/v1) - deployments.apps, statefulsets.apps, and so on.
  - Batch (batch/v1): jobs.batch, cronjobs.batch
  - networking.k8s.io/v1 - ingress.networking.k8s.io, networkpolicies.networking.k8s.io
- **resourceNameRegex** - matching resource name
- **labelSelector** - matching resource label
- **namespaces** - the namespace where the resource is available

3. The four kinds of patches that can be applied in transform CR.

JSON patch

A JSON patch supports operations like test, add, remove, and replace on specific paths. There are three fields need to define in the JSON patch:

- **op** - operation name
  - **test** - to test if specific value is present on the given path
  - **add** - to add the value on the given path
  - **remove** - to remove the value on the given path
  - **replace** - to replace the value on the given path
  - **copy** - to copy the value from path to the path
  - **move** - to move the value from path to the path
- **path** - on which the operation applies
- **value** - value on which these operations acted upon
- **from** - from which the value is copied

Note: For PVC, **copy** and **move** operation are supported for labels and annotations

For example, transform CR to transform multiple storage classes with JSON patch:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: Transform
metadata:
  name: transform-copy-move
  namespace: ibm-spectrum-fusion-ns
spec:
  transforms:
  - name: configmap-move-labels-to-annotations
    json:
    - op: move
      from: /metadata/labels/a
      path: /metadata/annotations/a
    subject:
      groupResource: configmaps
      labelSelector: c=d
  - name: configmap-move-annotations-to-labels
    json:
    - op: move
      from: /metadata/annotations/m
      path: /metadata/labels/m
    subject:
      groupResource: configmaps
      labelSelector: c=d
  - name: pvc-move-labels-to-annotations
    json:
    - op: move
      from: /metadata/labels/a
      path: /metadata/annotations/a
    subject:
      groupResource: persistentvolumeclaims
      labelSelector: c=d
  - name: pvc-move-annotations-to-labels
    json:
    - op: move
      from: /metadata/annotations/m
      path: /metadata/labels/m
    subject:
      groupResource: persistentvolumeclaims
      labelSelector: c=d
  - name: pvc-copy-labels-to-annotations
    json:
    - op: copy
      from: /metadata/labels/i
      path: /metadata/annotations/i
    subject:
      groupResource: persistentvolumeclaims
      labelSelector: i=j
  - name: pvc-copy-annotations-to-labels
    json:
    - op: copy
      from: /metadata/annotations/p
      path: /metadata/labels/p
    subject:
      groupResource: persistentvolumeclaims
      labelSelector: i=j
  - name: sts-copy-image
    json:
    - op: copy
      from: /spec/template/spec/containers/0/image
      path: /spec/template/spec/initContainers/0/image
    subject:
      groupResource: statefulsets.apps
      labelSelector: a=b
  - name: sts-copy-memory-request-to-limit
    json:
```

```
    - op: copy
      from: /spec/template/spec/containers/0/resources/requests/cpu
      path: /spec/template/spec/containers/0/resources/limits/cpu
    subject:
      groupResource: statefulsets.apps
      labelSelector: a=b
```

Note: `${SOURCE_STORAGE_CLASS_FS}`, `${TARGET_STORAGE_CLASS_FS}`, `${SOURCE_STORAGE_CLASS_RBD}`, `${TARGET_STORAGE_CLASS_RBD}`, these are the placeholders here which can be pass dynamically using `transformMapping` in restore CR.

Merge patch

Merges provided fields into the target resource using a simple JSON structure. A specific field needs to be defined.
patchData - provide a JSON structure

For example, transform CR to add labels to `statefulsets` using merge patch.

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: Transform
metadata:
  name: transform-using-merge-patch
  namespace: ibm-spectrum-fusion-ns
 - name: statefulsets-with-merge-patch-labels
    mergePatches:
    - patchData: |
        {
          "metadata": {
            "labels": {
              "modify-by": "${MODIFY_BY}"
            }
          }
        }
    subject:
      groupResource: statefulsets.apps
      labelSelector: 'a=b'
```

Note: `{MODIFY_BY}` is a placeholder here that can be pass dynamically using `transformMapping` in restore CR.

Strategic patch

Merges fields intelligently based on resource schema, preserving arrays like containers by name. A specific field needs to be defined.
patchData - provide a JSON structure

For example, transform CR to update image version and add a `env` variable using strategic patch.

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: Transform
metadata:
  name: transform-using-strategic-patch
  namespace: ibm-spectrum-fusion-ns
  - name: deployment-with-strategic-patch
    strategicPatches:
    - patchData: |
        {
          "spec":
            {
              "template":
                {
                  "spec": {
                    "containers": [
                      {
                        "name": "nginx-main",
                        "image": "nginx:1.25.6",
                        "imagePullPolicy": "Always",
                        "env": [
                          {
                            "name": "test",
                            "value": "${MODIFY_BY}"
                          },
                        ]
                      }
                    ]
                  }
                }
            }
        }
    subject:
      groupResource: deployments.apps
      resourceNameRegex: nginx-multi-container
```

String patch

Replaces specific strings in a target resource using key-value pairs. Two specific fields needs to be defined.

- **key** - fully qualified path where string need to be replaced
- **replaceStrings**
  - `Old` - old string, which needs to be replaced
  - `New` - new string, which needs to be replaced on the place of old string.

For example, transform CR to transform data in the configmap using string patch.

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: Transform
metadata:
  name: transform-using-string-patch
  namespace: ibm-spectrum-fusion-ns
spec:
  transforms:
```

```
  - name: configmap-with-string-patch
    stringPatch:
    - key: data/instance.json
      replaceStrings:
      - new: '\"persistence.cachingpv.storageClass\": \"${TARGET_STORAGE_CLASS_FS}\"'
        old: '\"persistence.cachingpv.storageClass\": \"transform\"'
    subject:
      groupResource: configmaps
      resourceNameRegex: dv-instance-config
```

4. After the transform CR is define, then create a restore CR with its reference.
   For example:

```
apiVersion: data-protection.isf.ibm.com/v1alpha1
kind: Restore
metadata:
  name: restore-with-transform
  namespace: ibm-spectrum-fusion-ns
spec:
  backup: <backup-name>
  targetCluster: <target-cluster-name>
  transform: transform-storage-classes
  transformMapping:
    INGRESS_HOST:  transformed.com
    SOURCE_STORAGE_CLASS_FS: ocs-storagecluster-cephfs
    TARGET_STORAGE_CLASS_FS: ocs-storagecluster-cephfs-1
    SOURCE_STORAGE_CLASS_RBD: ocs-storagecluster-ceph-rbd
    TARGET_STORAGE_CLASS_RBD: ocs-storagecluster-ceph-rbd-1
    MODIFY_BY: fusion-transform
```

   Define the two new fields `transform` and `transformMapping` in the restore CR.
   - `transform` - name of the transform CR.
   - `transformMapping` - If required, pass the variable that can be substituted in the transform CR.
5. Apply the restore CR.
   Note:
   - Transform CR should be present on the target cluster where restoring the application.
   - For `persistentvolumeclaims`, only JSON patch is supported, and all the 4 patches are supported for resources.
   - The test operation in JSON patch is used in conjunction with other operations (**add, replace, remove**). If the test operation **fails**, the entire patch is aborted and none of the subsequent operations are run in that transform rule.
   - Strategic Merge patch operates based on the name field within arrays (such as containers):
     - If the name field is missing, then the merge is skipped.
     - If a matching name field is present, then the existing entry is merged.
     - If a matching name field is not present, then a new entry is added to the array.
     - If a label key contains a slash (/), replace it with a (~) for it to be valid.
   - For more examples of using Transform CR, see Sample Transform CR.

   Transformation with custom recipe:
   Note: With default recipe, this setting is not applicable.
   To use the transformation with custom recipe, you need to add the following field in the group of the recipe

```
"disableTransform: false"
```

   For example:

```
apiVersion: spp-data-protection.isf.ibm.com/v1alpha1
kind: Recipe
metadata:
  name: db2-prod-backup-recipe
  namespace: ibm-spectrum-fusion-ns
spec:
  appType: db2
  groups:
    - includedNamespaces:
        - db2-prod
      name: db2-volumes
      type: volume
    - includedNamespaces:
        - db2-prod
      includedResourceTypes:
        - configmaps
        - secrets
        - services
        - serviceaccounts
      name: db2-resources
      type: resource
    - includedNamespaces:
        - db2-prod
      includedResourceTypes:
        - statefulsets.apps
      name: db2-statefulsets
      type: resource
...
```

   To enable the transformation, include `disableTransform: false` for this group `db2-statefulsets`.

```
- includedNamespaces:
        - db2-prod
      includedResourceTypes:
        - statefulsets.apps
      name: db2-statefulsets
      type: resource
      disableTransform: false
```

Backup and restore of IBM Software Hub to the same or different cluster

IBM Software Hub is integrated with IBM Fusion , which enables you to create online backups and to restore them to the same cluster or to a different cluster.

Note: IBM Software Hub supports backup and restore on both x86 and Power.
To create IBM Software Hub backups with IBM Fusion , see Creating and scheduling online backups of IBM Software Hub with IBM Fusion.

To restore IBM Software Hub online backup and restore to the same cluster with IBM Fusion , see IBM Software Hub online backup and restore to the same cluster with IBM Fusion.

To restore IBM Software Hub backups to a different cluster with IBM Fusion , see IBM Software Hub online backup and restore to a different cluster with IBM Fusion.

Remember: IBM Cloud Paks instance backups cannot be created when continuous backup is enabled on the MongoDB service.
Note: You can restore backup data to a different StorageClass provisioner on an alternate cluster, for example, restore data that was originally backed-up from a CephFS provisioned StorageClass to a Scale provisioned StorageClass.
The Restore CR includes an optional **`spec`** field, **`targetStorageClass`** where you can specify a specific storage class for the restore. It is only available through manually created Restore CRs from the command line.

- If the **`targetStorageClass`** is not provided, the same storage class as the backup is attempted for the restore.
- If the **`targetStorageClass`** field is not provided and the same storage class as the backup is not available, the default storage class of the system is used.

The **`targetStorageClass`** is not supported for IBM Software Hub.

IBM Fusion repository for recipes

The IBM Fusion public GitHub repository provides helpful utilities for IBM Fusion and allows members of the broader IBM Fusion community to share and contribute tools and knowledge.

This repository hosts the following categories of artifacts that can be used with IBM Fusion:

- Multiple application backup and restore recipes using IBM Fusion. It includes recipes for applications such as IBM Db2, PostgreSQL, Postgres EnterpriseDB, MongoDB, Redis, and others. Check the Recipes repository for the complete inventory of applications.
- Maximo Core and Manage application recipes. For the IBM Fusion GitHub repository, see Maximo Fusion recipes.
- Helpful utilities and scripts to explore backup and restore
  For example, you can run getRecipeWorkflow.sh scripts from the command line to view recipe workflow logs and run script getResources.sh to get complete list of resources backed up or restored.

If you like to develop recipes for other popular applications, read the following blog series on creating backup and restore recipes:

1. How IBM Fusion Simplifies Backup and Recovery of Complex OpenShift Applications - Part 1 "So Why is this Complex?"
2. How IBM Fusion Simplifies Backup and Recovery of Complex OpenShift Applications - Part 2 "Orchestrations and Recipes"
3. How IBM Fusion Simplifies Backup and Recovery of Complex OpenShift Applications - Part 3 “A Deeper Dive into Recipes"

Also, for more recipes, see Orchestrate a backup or restore.

Performance and scaling

Backup and restore performance depends on agent concurrency, data mover pod counts, and storage throughput. Tuning these parameters helps IBM Fusion meet capacity and latency requirements for large workloads.

- **Backup & restore resource self-monitoring**
  The imposed load on the Backup & restore service has a direct correlation to the number of concurrent Backup & restore jobs in progress. While the service can catch up from periodic spikes in activity, a sustained elevated level of activity may cause any one Backup & restore microservice to fall behind in job processing, which may result in job failures.
- **Backup & restore hub performance and scaling**
  Use this topic to understand the number of spokes you can scale up to and the number of spoke clusters that can be connected to a single hub. It outlines the capacities and the scalability considerations to ensure efficient backup and restore operations.
- **Backup & restore configuration parameters**
  Configuration parameters allow customization of IBM Fusion Backup & restore settings.

Backup & restore resource self-monitoring

The imposed load on the Backup & restore service has a direct correlation to the number of concurrent Backup & restore jobs in progress. While the service can catch up from periodic spikes in activity, a sustained elevated level of activity may cause any one Backup & restore microservice to fall behind in job processing, which may result in job failures.

The **Backup and Restore hub microservices**—such as `backup-service`, `job-manager`, `application-service`, `backup-policy` and `backup-location`—now include built-in resource monitoring for Memory and CPU consumption. This monitoring capability helps identify potential resource shortages by providing a Warning alert through IBM Fusion events. This alert notifies the user of a sustained level of activity the current microservice cannot sustain and recommends scaling out the microservice with another replica to ensure successful Backup & restore job operations.

Resource Usage Monitoring and Thresholds

When Memory or CPU usage exceeds a defined threshold (default is 85%) for a specific duration (default is 5 minutes), a IBM Fusion event with a warning severity is triggered to notify about the increased resource usage. For instance, if the Memory or CPU usage for the `backup-service` pod crosses the threshold, the following event gets generated:

```
Pod backup-service-xxx in namespace ibm-backup-restore exceeded 85% of memory/CPU limit for 5 minutes.
Consider scaling the deployment with an additional replica.
```

These alerts are marked as fixed after the Memory or CPU consumption falls behind the threshold limit.

To scale out with another replica, use the following command to get the existing replica count for the deployment from the 'AVAILABLE' column:

```
oc get deployment -n <backup-restore-namespace> <deployment-name>
```

Then perform the following command with an increase of 1 to the existing replica count:

```
oc scale deployment -n <backup-restore-namespace> <deployment-name> --replicas=<new-replica-count>
```

Configurable parameters

You can customize the threshold levels and monitoring duration by adding or modifying parameters in the `guardian-configmap`. Here are the parameters available for configuration:

memoryThresholdPercentage
: Sets the memory usage threshold as a percentage. Default is "85" for 85%.

cpuThresholdPercentage
: Sets the CPU usage threshold as a percentage. Default is "85" for 85%.

monitorDurationInSeconds
: Defines the duration (in seconds) that usage must exceed the threshold to trigger an alert. Default is 300 seconds (5 minutes).

Example configuration:

```
memoryThresholdPercentage: '85'
cpuThresholdPercentage: '85'
monitorDurationInSeconds: '300'
```

Backup & restore hub performance and scaling

Use this topic to understand the number of spokes you can scale up to and the number of spoke clusters that can be connected to a single hub. It outlines the capacities and the scalability considerations to ensure efficient backup and restore operations.

The Backup & restore hub service is efficiently designed to handle large-scale backup and restore concurrent jobs across clusters. With proven scalability and performance. The system is tested successfully to handle up to 1000 concurrent jobs. It serves as a reference point and not a limitation; the hub can scale further with appropriate resource allocation.

Sizing blueprints for varying workloads

The following table provides the recommended configurations for each required service when handling different numbers of concurrent jobs. As the number of concurrent jobs increases, you can scale out (increase the replicas) or scale up (increase the resource limits) for the required services as outlined:

Note:

- Services not listed in the blueprint table do not require scaling. These services can handle the load without modification because they do not have scaling requirements for up to 1000 concurrent jobs.
- Some services, such as `mongodb-v2` or `operator-control-manager` pods in the `ibm-backup-restore` namespace, cannot be scaled out by increasing replicas. For these services, increase the CPU and memory resources to ensure optimal performance as the number of concurrent jobs increases.
- The `guardian-bridge` service automatically scales to handle the additional load and provides backward compatibility when connected spokes are at a version earlier than IBM Fusion 2.13.

Table 1. Sizing blueprints

Clusters	Concurrent Jobs	Mongodb-v2				Kafka broker pool	Backup service	Backup policy service	Application service	Job manager service	Vi
		Replicas	CPU	Memory	Ephemeral	Replicas	Replicas	Replicas	Replicas	Replicas	

100	1000	3	1	1 GiB	0.5 GiB	3	1	1	1	1	4
200	2000	3	2	1 GiB	0.5 GiB	3	1	1	1	1	4
300	3000	3	3	1.5 GiB	1 GiB	6	1	1	1	1	5
400	4000	3	3.5	1.5 GiB	1 GiB	6	1	1	1	1	5
500	5000	3	4	2 GiB	1.5 GiB	9	1	1	1	1	6
600	6000	3	4.5	2 GiB	1.5 GiB	9	1	1	1	1	6
700	7000	3	5	2 GiB	1.5 GiB	12	1	1	1	1	7
800	8000	3	6	2.5 GiB	1.75 GiB	12	2	1	1	2	8
900	9000	3	7	2.5 GiB	1.75 GiB	15	2	2	2	2	9
1000	10000	3	8	3 GiB	2 GiB	15	3	2	2	2	10

Scaling Services

Use the following commands to scale out the desired replicas or scale up resource limits for the required services:

mongodb-v2 - scale up resource limits

The mongodb-v2 `StatefulSet` cannot be scaled out. Increase CPU, memory, and ephemeral storage limits as the workload grows:

```
oc set resources sts mongodb-v2 --limits=cpu=<desired-cpu>,memory=<desired-memory>,ephemeral-storage=<desired-ephemeral> --containers=mongodb -n <backup-restore-namespace>
```

For example:

```
oc set resources sts mongodb-v2 --limits=cpu=8,memory=3Gi,ephemeral-storage=2Gi --containers=mongodb -n ibm-backup-restore
```

Kafka broker-pool - scale out replicas

The Kafka broker-pool is managed as a `KafkaNodePool` resource. Scale its replicas using:

```
oc patch kafkanodepool broker-pool -n <backup-restore-namespace> --type='json' -p='[{"op": "replace", "path": "/spec/replicas", "value": <desired-replicas>}]'
```

For example:

```
oc patch kafkanodepool broker-pool -n ibm-backup-restore --type='json' -p='[{"op": "replace", "path": "/spec/replicas", "value": 15}]'
```

`backup-service` - scale out replicas

```
oc scale deployment backup-service --replicas=<desired-replicas> -n <backup-restore-namespace>
```

For example with 100 clusters/1000 jobs (3 replicas):

```
oc scale deployment backup-service --replicas=3 -n ibm-backup-restore
```

`backuppolicy-deployment` - scale out replicas

```
oc scale deployment backuppolicy-deployment --replicas=<desired-replicas> -n <backup-restore-namespace>
```

For example with 90-100 clusters (2 replicas):

```
oc scale deployment backuppolicy-deployment --replicas=2 -n ibm-backup-restore
```

`applicationsvc` - scale out replicas

```
oc scale deployment applicationsvc --replicas=<desired-replicas> -n <backup-restore-namespace>
```

For example with 80-100 clusters (2 replicas):

```
oc scale deployment applicationsvc --replicas=2 -n ibm-backup-restore
```

`job-manager` - scale out replicas

```
oc scale deployment job-manager --replicas=<desired-replicas> -n <backup-restore-namespace>
```

For example with 80-100 clusters (2 replicas):

```
oc scale deployment job-manager --replicas=2 -n ibm-backup-restore
```

`windowEndTime` - update through `BackupPolicy` CR

The `windowEndTime` is a field under `spec.schedule` in the `BackupPolicy` CR. The value is in HH:MM format.
Note: `windowEndTime` must be set to a minimum of 4 hours later than the scheduled (cron) start time. This can also be updated from the IBM Fusion user interface by navigating to Backup & Restore > Policies and by editing the required policy.
Update the `windowEndTime` field (HH:MM format) under `spec.schedule`. Ensure the value is at least 4 hours after the policy cron schedule time:

```
oc patch fbp <policy-name> -n <ibm-spectrum-fusion-ns> --type='json' -
p='[{"op":"replace","path":"/spec/schedule/windowEndTime","value":"<HH:MM>"}]'
```

For example with 100 clusters/1000 jobs (window ending at 10:00):

```
oc patch fbp <policy-name> -n <ibm-spectrum-fusion-ns> --type='json' -
p='[{"op":"replace","path":"/spec/schedule/windowEndTime","value":"10:00"}]'
```

`cancelJobAfter` - update in job-manager deployment

The `cancelJobAfter` value is specified in milliseconds and is configured as an environment variable in the job-manager deployment.
Patch the `env` variable value (index 3 corresponds to `cancelJobAfter`):

```
oc patch deployment job-manager -n <backup-restore-namespace> --type='json' -
p='[{"op":"replace","path":"/spec/template/spec/containers/0/env/3/value","value":"<milliseconds>"}]'
```

For example with 80 clusters/800 jobs (10800000 ms = 3 hours):

```
oc patch deployment job-manager -n ibm-backup-restore --type='json' -
p='[{"op":"replace","path":"/spec/template/spec/containers/0/env/3/value","value":"10800000"}]
```

`MaxConcurrentReconciles` - update in `dp-operator` CSV

`MaxConcurrentReconciles` is configured as an environment variable in the `guardian-dp-operator` cluster service version.
Patch the value (index 3 corresponds to `MaxConcurrentReconciles` based on the current CSV):

```
oc patch csv <guardian-dp-operator-csv-name> -n <backup-restore-namespace> --type='json' -p='[{"op": "replace", "path": "/spec/install/spec/deployments/0/spec/template/spec/containers/1/env/3/value", "value": "<desired-value>"}]'
```

For example with 80-100 clusters (20 reconciles):

```
oc patch csv guardian-dp-operator.v2.13.0-20260402145657 -n ibm-backup-restore --type='json' -p='[{"op": "replace",
"path": "/spec/install/spec/deployments/0/spec/template/spec/containers/1/env/3/value", "value": "20"}]'
```

Backup & restore configuration parameters

Configuration parameters allow customization of IBM Fusion Backup & restore settings.

Change defaults for IBM Fusion Backup & restore agent

deleteBackupWait
: Timeout for restic command to delete backup in S3 storage. Set in minutes. The default value is 20 minutes, and the allowed range is 10 to 120.

pvcSnapshotMaxParallel
: Number of threads available to take concurrent snapshots. The default value is 20.

backupDatamoverTimeout
: Maximum amount of time in minutes for the datamover to complete backup. The default value is 1200 seconds, and the allowed range is 10 to 14400. After you modify backupDatamoverTimeout, update cancelJobAfter.

restoreDatamoverTimeout
: Maximum amount of time in minutes for datamover to complete restore. The default value is 1200 seconds, and the allowed range is 10 to 14400. After you modify restoreDatamoverTimeout, update cancelJobAfter.

snapshotRestoreJobTimeLimit
: This parameter is not used.

pvcSnapshotRestoreTimeout
: Timeout for creating PVC from snapshot in minutes. The default value is 15 minutes.

kafka-thread-size
: Controls the number of concurrent backup or restore jobs processed by a transaction manager pod. The default value is 10.

snapshotTimeout
: Timeout for snapshot to resolve to the ready state in minutes. The default value is 20, and the allowed range is 10 to 120.

datamoverJobpodEphemeralStorageLimit
: Datamover pod ephemeral storage limit. The default value is 2000Mi.

datamoverJobPodDataMinGB
: Minimum PVC capacity for each datamover pod before a new datamover pod is started. It is set in GB, and the default value is 10 GB.

datamoverJobpodMemoryLimit
: It is the Datamover pod memory limit, and the default value is 15000Mi.

datamoverJobpodCPULimit
: It is the Datamover pod CPU limit, and the default value is 2.

cancelJobAfter
: If you modify backupDatamoverTimeout or restoreDatamoverTimeout, update the job-manager deployment configuration parameter cancelJobAfter. It is the maximum amount of time in milliseconds that the job-manager waits before it cancels the long-running job. The default value is 3600000 (1 hour).

MaxNumJobPods
: Implement configurable MaxNumJobPods for datamovers.
This field helps to control the number of PersistentVolumeClaims that is attached to datamover pods during backup and restore to the **`BackupStorageLocation`**. It is set on a per install basis. If you have spoke clusters installed in your IBM Fusion installation, then set on each spoke cluster individually. It is not a global field that applies to all clusters in the installation.

  It helps to distribute the storage load across multiple nodes of the cluster when available. Some StorageClasses impose a maximum number of PVCs that can be attached to an individual node of the cluster. This field helps to manage this StorageClass limitation. To find out whether your StorageClass has this limitation, check whether the CSINode of your storage provider has the spec.drivers[].allocatable.count field set. The VPC Block on IBM Cloud is one such storage provider with this limitation, typically 10 per node. If you increase the number of pods and the application has more than 30 PVCs, it decreases the number of PVCs attached to each node. If it goes below this, you can use the default, which is more than sufficient.

  This field can increase or decrease the maximum number of datamover pods that are assigned in each backup or restore. More pods use more resources such as CPU and memory, and can help improve performance for backups and restores with larger numbers of PVCs. Increasing this value may help if the number of PVCs to be backed up or restored at the same time is more than 30.

  This field does not guarantee the creation of more datamover pods. A number of heuristics are used at runtime to help determine the assignment of PVCs to datamovers, including total PVC capacity, number of PVCs, amount of data transferred during previous backups, total number of PVCs handled, storage providers involved, among others. This field changes the maximum allowed, and it does not guarantee the specified number of datamovers.

  For more information about how to change the value, see Change defaults for IBM Fusion Backup & restore agent.

DeleteBackupRequest CR cleanup
: The **`DeleteBackupRequest`** CRs in the Completed state gets automatically deleted after a default retention of 14 days. You can set the **`dbrCRRetention`** configuration parameter in ConfigMap guardian-configmap to change the default. By default, the cleanup thread runs every 2 hours and cleans up a maximum of 100 **`DeleteBackupRequest`** CRs in one run. These two values can be overridden by specifying **`dbrCleanupCheckInterval`** and **`dbrCleanupBatchSize`** configuration parameters in the **`guardian-configmap`** ConfigMap.

dbrCRRetention
: Number of days after which **`DeleteBackupRequest`** CRs in the Completed state get deleted. If not specified, then 14 days is the default.

dbrCleanupCheckInterval
: Interval, in hours, at which the `DeleteBackupRequest` cleanup thread runs. If not specified, 2 hours is the default.

dbrCleanupBatchSize
: Number of `DeleteBackupRequest` CRs to cleanup in one run of the cleanup thread. If not specified, 100 is the default.

Change defaults for IBM Fusion Backup & restore agent

Do the following steps to change the value:

1. In the OpenShift® Container Platform console, click Operators > Installed Operators.
2. Change the project to the IBM Spectrum Fusion Backup and Restore namespace. For example `ibm-backup-restore`.
3. Click to open IBM Fusion Backup & Restore Agent.
4. Click the Data Protection Agent tab and click the dpagent install. Alternatively, if you want to use the OC command:

   ```
   oc edit -n ibm-backup-restore dataprotectionagent
   ```

5. Go to the YAML tab.
6. Edit spec.transactionManager.datamoverJobPodCountLimit. The value must be numeric and in quotes. For example, '3', '5', '10'

Backup and restore large number of files

When you back up and restore a large number of files located on CephFS, you must perform the following additional steps for the operations to succeed. The optimal values depend on your individual environment, but the following values are representative for a backup and restore of a million files.

- Prevent the transaction manager from failing long running backup jobs. In the `ibm-backup-restore` project, edit the config map named `guardian-configmap`. Look for `backupDatamoverTimeout`. This value is in minutes, and the default is 20 minutes. For example, increase this value to 8 hours (480).
- Prevent the job manager from canceling long running jobs. In the `ibm-backup-restore` project, edit the `job-manager` deployment. Under `env`, look for `cancelJobAfter`. This value is in milliseconds, and the default is 1 hour. For example, increase this value to 20 hours (72000000).
- Prevent the transaction manager from failing long running restore jobs. In the `ibm-backup-restore` project, edit the config map named `guardian-configmap`. Look for `restoreDatamoverTimeout`. This value is in minutes, and the default is 20 minutes. For example, increase this value to 20 hours (1200).
- In the same config map, increase the amount of ephemeral storage the data mover is allowed to use by increasing datamoverJobpodEphemeralStorageLimit to 4000Mi or more.
- OpenShift Data Foundation parameters.
  - Increase the resources available to OpenShift Data Foundation. Increase the limits and requests for the two MDS pods, a and b, for example to 2 CPU and 32 Gi memory. For more information about the changes, see Changing resources for the OpenShift Data Foundation components.
  - Prevent SELinux relabelling
    At restore time, OpenShift will attempt to relabel each of the files. If it takes too long, the restored pod will fail with CreateContainerError. This article explains the situation and some of the possible workarounds to prevent the relabeling:https://access.redhat.com/solutions/6221251.

Additional parameters are available when you backup and restore large number of files that are located on CephFS. The optimal values depend on your individual environment, but the values in this example represent backup and restore of a million files. For such large number of files, you must be on OpenShift Container Platform 4.14 or later, and Data Foundation 4.12 or later.

Set up the following Backup & Restore parameters:

- In the same config map, increase the amount of ephemeral storage the data mover is allowed to use. Increase `datamoverJobpodEphemeralStorageLimit` to 4000Mi or more.

Set up the following Red Hat® OpenShift Data Foundation parameters:

- Increase the resources available to Red Hat OpenShift Data Foundation. Increase the limits and requests for the two MDS pods, for example, to 2 cpu and 32 Gi memory. For the procedure to change, see Changing resources for the OpenShift Data Foundation components.
- Set up to Prevent SELinux relabeling:
  At restore time, OpenShift attempts to relabel each of the files. If it takes too long, the restored pod fail with `CreateContainerError`. This article explains the situation and some of the possible workarounds to prevent the relabeling: https://access.redhat.com/solutions/6221251.

Use the following steps to understand whether your backups fail due to a large number of files:

1. Run the following command to check whether the MDS pods are restarting:

   ```
   oc get pod -n openshift-storage |grep mds
   ```

2. If they are restart, check the termination reason:
   a. Describe the pod.
   b. Check whether the termination is `OOM Kill`.
3. Run the following command to check the memory usage by the MDS pods and monitor for memory usage:

   ```
   oc adm top pod -n openshift-storage
   ```

4. If the memory usage keeps spiking until the pod restarts, then see Changing resources for the OpenShift Data Foundation components.

Backup and restore large number of resources

When you back up and restore a large number of cluster resources, the velero pod resource limits must be increased for the operations to succeed. The optimal values depend on your individual environment, but the following values are representative for backup and restore of 1500 resources.

```
oc patch dpa velero -n <backup-restore-namespace> --type merge -p '{"spec": {"configuration": {"velero": {"podConfig":
{"resourceAllocations": {"limits": {"ephemeral-storage": "1Gi"}}}}}}}'
```

```
oc patch dpa velero -n <backup-restore-namespace> --type merge -p '{"spec": {"configuration": {"velero": {"podConfig":
{"resourceAllocations": {"limits": {"memory": "2Gi"}}}}}}}'
```

Configure DataMover type for Backup and Restore

The DataMover type dictates both the entity responsible for backing up data on PersistentVolumeClaims and the method of storage. **kopia** is the only allowed value.

- The default value is **kopia**.
- The **legacy** DataMover for versions equal to or lower than 2.8

The value can be set globally and per PolicyAssignment. The two different DataMover types are not cross-compatible. Backups with one DataMover type use that type during restore. Existing backups continue to expire normally per their Policy.

Change the DataMover type used across all new backups

From the OpenShift console:

1. Log in to OpenShift Console.
2. Go to Workloads > ConfigMaps.
3. Select the IBM Fusion install namespace.
4. Open **isf-data-protection-config** and update **data.Datamover** field.

Example command:

```
oc patch configmap -n ibm-spectrum-fusion-ns isf-data-protection-config -p '{"data":{"DataMover":"kopia"}}'
```

Here, replace namespace **ibm-spectrum-fusion-ns** with your Fusion namespace. Also, this example shows **kopia** DataMover. For **legacy** DataMover, replace **kopia** to **legacy**.

Choose DataMover during policy assignment

To facilitate a trial of the DataMover options, select it by annotating PolicyAssignment objects. Each PolicyAssignment object associates an existing Policy and BackupStorageLocation with an application to backup. By setting this value according to the following instructions, the global setting gets ignored. Consequently, all new backups associated with the PolicyAssignment use the specified DataMover type in the annotation.

1. Search by label to find the existing PolicyAssignment for your applications.
   - To search by application - **dp.isf.ibm.com/application-name=<\application name>**
   - To search by policy - **dp.isf.ibm.com/backuppolicy-name=<\policy name>**
   - To search by backpropagation - **dp.isf.ibm.com/backupstoragelocation-name=<\backupstoragelocation name>**
     The general naming format is as follows:

     <application name>-<\policy name>-<cluster url>

     Example:

     aws-20240716-220437-awsdaily-apps.bnr-hcp-munch.apps.blazehub01.mydomain.com
2. Change the value of DataMover.
   From the OpenShift console:
   a. Go to Operators > Installed Operators.
   b. Select the IBM Fusion install namespace.
   c. Open IBM Fusion.
   d. Go to the Policy Assignment tab and change the DataMover value.
   Using commands:
   a. Get the policy assignment object:

      oc get policyassignment -n ibm-spectrum-fusion-ns

   b. Add the annotation to the PolicyAssignment object.

      dp.isf.ibm.com/dataMover: <datamover type>

      Example:

      dp.isf.ibm.com/dataMover: legacy

Choose nodes for DataMovers for OADP DataMover

From 2.9 version onwards, the Backup & restore is transitioning from the in-house DataMover to OADP. DataMover controller using restic as the data transfer software to OADP, which uses its own implementation of DataMover and controller with kopia.

Path to locate the custom resource

On Backup & restore namespace, do the following steps:

1. Go to Install Operators > OADP Operator.
2. On the Operator details page, select DataProtectionApplication tab and velero instance.
3. Select YAML tab to update configuration parameters.

spec.datamoverConfiguration

The field **spec.datamoverConfiguration** is used to configure the new OADP DataMovers. For backwards compatibility, you can continue to use the fields under **spec.transactionManager** to configure the DataMovers from the previous releases.

spec.datamoverConfiguration.nodeAgentConfig.env

This array specifies environment variables inserted into the node-agent pods. The example value of HOME allows setting the folder used by kopia for local files. It controls the on-container folder where cache files related to the storage repository are stored. The default value is `/home/velero` and it uses local ephemeral storage.

```
spec:
  datamoverConfiguration:
    nodeAgentConfig:
```

```
      env:
      - name: HOME
      value: /home/velero
```

spec.datamoverConfiguration.nodeAgentConfig.labels
: This field adds additional labels to the node-agent daemonset and pods. It is an optional field to organize pods and does not affect backup and restore behavior.This field is a YAML object.

```
spec:
 datamoverConfiguration:
   nodeAgentConfig:
     labels:
       cloudpakbackup: datamover
```

spec.datamoverConfiguration.nodeAgentConfig.tolerations
: This is an array field that specifies the conditions for nodes that permit the execution of Pods in resource-constrained environments or multi-architecture clusters. For more information about this field, see Kubernetes documentation.

```
spec:
 datamoverConfiguration:
   nodeAgentConfig:
     tolerations:
     - key: "kubernetes.io/arch"
       operator: "Equal"
       effect: "amd64"
```

spec.datamoverConfiguration.nodeAgentConfig.resourceAllocations
: This field defines the resource allocation for the node-agent controller, including both scheduling requirements and limits.
If backups or restores are causing evictions in the node-agent pods, increase the value of this field. Go through the Events in the Backup and Restore install namespace to check resource violations:

  - The status field of the Daemonset node-agent
  - The OpenShift Dashboard for resource monitoring in the Backup and Restore namespace
    See OpenShift documentation.
  - Grafana in the Backup and Restore namespace if installed
    See OpenShift documentation
  - Other applications that use Prometheus metrics.

  Example:

```
spec:
 datamoverConfiguration:
   nodeAgentConfig:
       resourceAllocations:
         limits:
           cpu: "2"
           ephemeral-storage: "50Mi"
           memory: "2048Mi"
         requests:
           cpu: "200m"
           ephemeral-storage: "25Mi"
           memory: "256Mi"
```

spec.configuration.nodeAgent.podConfig.nodeSelector
: This field decides the nodes where the DaemonSet node agent must run. This feature is a crucial security feature to restrict the HostPath mount of the node-agent that exposes the following values:

  - All PersistentVolumeClaims
  - Projected API volumes such as configmaps and secrets through volume mount
  - Kubernetes API tokens that are in use on the node

  Note: As this field causes Restore failures, remove it before restore operations.
  If needed, this field can isolate the Daemonset node-agent Pods to Nodes without any security concerns.

  The pods are assigned to Nodes that match the labels attached to the Nodes. If more than one label is used, the labels are treated in a logical AND. Only Nodes that match all of the labels runs the Pod.

  For example, if you set the nodeSelector to "kubernetes.io/hostname: bnr-hcp-munch-6df6702b-2bv7z", then the Daemonset node-agent Pods run only the node with this label.If there is only one node with this label, Daemonset node-agent is restricted to deploying Pods to only the labeled node.

```
spec:
   configuration:
      nodeAgent:
         podConfig:
            nodeSelector:
               kubernetes.io/hostname: bnr-hcp-munch-6df6702b-2bv7z
```

spec.datamoverConfiguration.datamoverPodConfig
: This setting manages the PVCs, resource allocations, and the location within the cluster where the datamover pods operate. The LoadConcurrency provides the number and LoadAffinity provides the location.

spec.datamoverConfiguration.datamoverPodConfig.loadConcurrency
: This field controls the maximum number of datamovers assigned to nodes in the cluster.
If there is anything that appears to be in conflict, see Velero documentation.

This setting controls the number of datamovers deployed per node-agent controller pod, and it consists of two parts:

- **`spec.datamoverConfiguration.datamoverPodConfig.loadConcurrency.globalConfig`**
  This sets the maximum number of datamovers deployable from a single node-agent controller pod for a generic node in the cluster. The subsequent field overrides globalConfig and manages this setting independently. It is of type integer with default value of five and minimum value of 1. If loadConcurrency is added to the config, this value is REQUIRED.

- **`spec.datamoverConfiguration.datamoverPodConfig.loadConcurrency.perNodeConfig`** is optional.
  This allows setting a maximum number of datamovers assigned to a node. Example:

```
loadConcurrency:
  globalConfig: 2
  perNodeConfig:
  - nodeSelector:
      matchLabels:
        kubernetes.io/hostname: node1
    number: 3
  - nodeSelector:
      matchLabels:
        beta.kubernetes.io/instance-type: Standard_B4ms
    number: 5
```

  The perNodeConfig uses the default nodeSelector elements from Kubernetes to select nodes by using labels. When multiple labels are specified in a matchLabels, the selected nodes are found using an AND operation, meaning all the labels must be found. These also take a more complex matchExpressions field as an alternative to nodeSelector. For more information about assigning pods to nodes, see Kubernetes documentation.
  In this example, the node with a label **`kubernetes.io/hostname=node1`** can run up to 3 DataMovers at once. And a node with label **`beta.kubernetes.io/instance-type=Standard_B4ms`** can run up to 5 DataMovers. With the globalConfig set to 2, all other nodes with matching storage to the PVC may run up to 2 DataMovers at the same time.

spec.datamoverConfiguration.datamoverConfiguration.datamoverPodConfig.LoadAffinity
: This field controls the nodes in the cluster where the DataMovers must run. For more information, see Velero reference documentation.

```
loadAffinity:
- nodeSelector:
    matchLabels:
      beta.kubernetes.io/instance-type: Standard_B4ms
    matchExpressions:
    - key: kubernetes.io/hostname
      values:
      - node-1
      - node-2
      - node-3
      operator: In
    - key: xxx/critial-workload
      operator: DoesNotExist
```

The value is a list of **`nodeSelectors`**. The selected nodes are controlled through a nodeSelector field that matches the labels. Each member of the list is evaluated independently. The nodes that match the element are combined to create the final list where the DataMovers can run. If the **`loadConcurrency.globalConfig`** is set to 0, then only the selected nodes in **`loadAffinity`** and **`perNodeConfig`** can run the backup and restore jobs.

Ensure that you meet the following criteria.

- The intersection of nodes matching loadConcurrency.perNodeConfig and loadConcurrency.loadAffinity is where the jobs run. Ensure the size of the intersection is at least one node.
- If the storage used during the backup or restore process is only available on a subset of nodes and **`loadConcurrency.globalConfig`** is set to 0, you must select at least one node where the storage is accessible. Otherwise, the job fails.
- If the **`loadConcurrency.globalConfig`** is set to 0 and the selected nodes do not have enough resources in CPU and memory to schedule a DataMover pod, then the backup or restore job fails.

spec.datamoverConfiguration.datamoverPodConfig.podResources
: Controls the amount of CPU and memory that is made available by the DataMovers. It has the following sub-fields and default values are set for the missing fields. Example:

```
podResources:
  cpuRequest: 2
  memoryRequest: 4Gi
  ephemeralStorageRequest: 5Gi
  cpuLimit: 4
  memoryLimit: 16Gi
  ephemeralStorageLimit: 5Gi
```

For more information about the default Velero resource profile for large clusters, see CPU and memory requirements in Red Hat documentation.

Resource limits must be sufficient to back up the most resource-intensive volumes in the cluster.

spec.datamoverConfiguration.datamoverPodConfig.backupPVC
: This section defines the format for the PersistentVolumeClaim used in the backup process. See Velero documentation.
During backup, each execution of a volumegroup in a Recipe (default Recipe is all PersistentVolumeClaims), a VolumeSnapshot gets created from a PersistentVolumeClaim representing the volume state at the time of the VolumeSnapshot timestamp. Some storage systems, such as CephFS from Data Foundation and IBM Storage Scale, implement shallow-copy backup volume exposure without file copy to address performance concerns compared to regular volume restores.

By default, the agent configures CephFS, IBM Storage Scale, and NFS PersistentVolumeClaims to be created in ReadOnlyMany mode during backups.

Setting values in **`backupPVC`** you can configure PersistentVolumeClaims from any StorageClass to use their "shallow-copy" implementation. This way, IBM Fusion does not need the specifics, and you can use alternative StorageClasses to access different storage features.

Example:

```
backupPVC:
  storage-class-1:
    storageClass: backupPVC-storage-class
    readOnly: true
  storage-class-2:
    storageClass: backupPVC-storage-class
  storage-class-3:
    readOnly: true
```

In this example, a data transfer of PersistentVolumeClaims of StorageClass "storage-class-1" makes use of an alternative StorageClass "backupPVC-storage-class" to create the volume with `accessModes: ReadOnlyMany`. The storage-class-2 is same as storage-class-1 except that the `accessModes` remains unchanged.

In storage-class-3, no alternate StorageClass can be used, and the volume gets created with `accessModes: ReadOnlyMany`. All DataMovers mount the backup volume as ReadOnly, and this affects the behavior of storage on volume creation. The original application `PersistentVolumeClaim` is not modified. Only the `PersistentVolumeClaim` used for the backup is modified.

If the StorageClass belongs to storage systems CephFS or IBM Storage Scale, the setting overwrites the agent's internal behavior with the one configured.

Not all storage systems support `PersistentVolumeClaims` with `accessModes: ReadOnlyMany`. See your appropriate storage documentation. The OpenShift does not provide a way to check what `accessModes` are supported prior to `PersistentVolumeClaim` creation.

To use PVC storage as a local cache for kopia datamovers:

1. Update ConfigMap node-agent-config.

   ```
   kind: ConfigMap
   apiVersion: v1
   metadata:
   data:
     node-agent-config: '{"loadConcurrency": {"globalConfig": 6}, "podResources": {"cpuRequest": "500m", "cpuLimit":
   "4", "memoryRequest": "500Mi", "memoryLimit": "4Gi", "ephemeralStorageRequest": "5Gi", "ephemeralStorageLimit":
   "5Gi"}, "backupPVC": {"ibm-spectrum-fusion-mgmt-sc": {"storageClass": "ibm-spectrum-fusion-mgmt-sc", "readOnly":
   true, "spcNoRelabeling": true}, "ocs-storagecluster-cephfs": {"storageClass": "ocs-storagecluster-cephfs",
   "readOnly": true, "spcNoRelabeling": true}}}'
   ```

2. Add the following field highlighted with ** ***:

   ```
   kind: ConfigMap
   apiVersion: v1
   metadata:
   data:
     node-agent-config: '{**"storage":{"storageClassName":"<storage class to use for PVC>","size":"<size of
   PVC>"}**,"loadConcurrency":{"globalConfig":6},"podResources":
   {"cpuRequest":"500m","cpuLimit":"4","memoryRequest":"500Mi","memoryLimit":"4Gi","ephemeralStorageRequest":"5Gi","ephe
   meralStorageLimit":"5Gi"},"backupPVC":{"ibm-spectrum-fusion-mgmt-sc":{"storageClass":"ibm-spectrum-fusion-mgmt-
   sc","readOnly":true,"spcNoRelabeling":true},"ocs-storagecluster-cephfs":{"storageClass":"ocs-storagecluster-
   cephfs","readOnly":true,"spcNoRelabeling":true}}}'
   ```

   Where the `storageClassName` is the `StorageClass` to allocate the storage from. Example: `ibm-spectrum-fusion-mgmt-sc`. The size is the Kubernetes resource size field. Examples: "500Mi" "5Gi" "10Gi" "20Gi" "50Gi".
3. Save the ConfigMap.
4. Restart the pods of `daemonset node-agent`.
5. Replace the namespace of `ibm-backup-restore` with the Fusion Backup & Restore install namespace.

   ```
   oc delete pod -n ibm-backup-restore -l name=node-agent
   ```

spec.datamoverConfiguration.datamoverPodConfig.maintenanceConfig

Kopia requires periodic maintenance jobs on the backup repository. See Velero documentation.
For each namespace, a Kopia repository is created and maintained. If a cloud object storage repository is used, automatic object expiration must not be used as it triggers repository corruption. The maintenance tasks enhance efficiency and reduce resource usage in future backup and restore jobs, typically lasting less than a minute. Their resource usage is similar to backup expiration tasks. These maintenance jobs run by default every 4 hours. The maintenance frequency can be adjusted by edited the `OADP DataProtectionAgents` under the field `spec.configuration.velero.args.default-repo-maintain-frequency`. These are the "Quick Maintenance Tasks" described by Kopia. See https://kopia.io/docs/advanced/maintenance/.

The "Full Maintenance Tasks" are run during backup expiration. The only available configuration item through the `DataProtectionAgent` object is the pod resources used during maintenance jobs.

spec.datamoverConfiguration.datamoverPodConfig.maintenanceConfig.resourceAllocations

This field is set using the Kubernetes `resourceAllocations` object format. https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
The following example is also the default values. Missing values such as `requests.cpu` are set to unlimited.

```
datamoverConfiguration:
    datamoverPodConfig:
      podResources:
        cpuLimit: '4'
        cpuRequest: '2'
        memoryLimit: 16Gi
        memoryRequest: 4Gi
        ephemeralStorageRequest: 5Gi
        ephemeralStorageLimit: 5Gi
```

The default is 5Gi.

spec.datamoverConfiguration.maintenanceConfig.jobSettings.ttl

This adds a Time To Live to Velero maintenance jobs. Velero defaults to keeping a record of 3 previously completed maintenance jobs. IBM Fusion changes this default value to 1. When you add this field, the Velero maintenance jobs are removed within seconds after completion.

```
  datamoverConfiguration
```

```
    maintenanceConfig:
      jobSettings:
        ttl: 240
      podResources:
        cpuLimit: '4'
        cpuRequest: '2'
        memoryLimit: '4Gi'
        memoryRequest: '500Mi'
        ephemeralStorageLimit: '5Gi'
        ephemeralStorageRequest: '5Gi'
```

If you do not set this field and you have `spec.datamoverConfiguration.maintenanceConfig.storage`, then it is automatically set to 300. For more information, see https://kubernetes.io/docs/concepts/workloads/controllers/ttlafterfinished/.

spec.datamoverConfiguration.maintenanceConfig.storage

This section enables you to use a PersistentVolumeClaim to act as a cache during Velero maintenance jobs. Most usage of ephemeral-storage during maintenance jobs originates from the kopia datamover cache. The addition of this field with a sufficiently sized PersistentVolumeClaim to handle the required cached data eliminates the aforementioned ephemeral-storage usage.

```
    maintenanceConfig:
      jobSettings:
        ttl: 240
      storage:
        storageClassName: "ocs-external-storagecluster-ceph-rbd"
        size: 25Gi
      podResources:
        cpuLimit: '4'
        cpuRequest: '2'
        memoryLimit: '4Gi'
        memoryRequest: '500Mi'
        ephemeralStorageLimit: '5Gi'
        ephemeralStorageRequest: '5Gi'
```

If storage is configured, then you need the following fields:

- `storageClassName`
  The name of the `StorageClass` to be used with the `PersistentVolumeClaim`. The `StorageClass` must support `accessMode: ReadWriteOnce`.
- `size`
  The size of the volume to be created. This field uses the existing Kubernetes resources request format such as "5Gi" and "250Mi".

  The size of the created PVC is to be used by each data mover pod.

During maintenance jobs, a PersistentVolumeClaim gets created. It is deleted when the job is removed based on the `spec.datamoverConfiguration.maintenanceConfig.jobSettings.ttl` field. If `spec.datamoverConfiguration.maintenanceConfig.jobSettings.ttl` is not specified and the storage configuration exists, then the ttl gets set to 300 (five minutes).

Advanced: Per Namespace Configuration

The maintenance config values mentioned previously are applied globally. These fields can also be applied manually by editing the appropriate `maintenance-config` ConfigMap. It allows overrides by namespace in case of resource usage differences.
The format is the same as `DataProtectionAgent spec.datamoverConfiguration.maintenanceConfiguration` in JSON instead of YAML.

The configuration is applied using a JSON format where the key is the repository. If a value is set for a repository, the value overrides the global values. For instance, if the global values are as described in the following example:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: maintenance-config
  namespace: ibm-backup-restore
data:
  global: '{"jobSettings":{"ttl":300},"podResources":
{"cpuRequest":"500m","cpuLimit":"4","memoryRequest":"500Mi","memoryLimit":"4Gi","ephemeralStorageLimit":"5Gi","ephemeralSt
orageRequest":"5Gi"}, "storage":{"storageClassName":"ocs-external-storagecluster-ceph-rbd","size":"20Gi"}}'
```

Additional keys are in the `<namespacename>-<bsl name>-kopia` format:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: maintenance-config
  namespace: ibm-backup-restore
data:
  global: '{"jobSettings":{"ttl":300},"podResources":
{"cpuRequest":"500m","cpuLimit":"4","memoryRequest":"500Mi","memoryLimit":"4Gi","ephemeralStorageLimit":"5Gi","ephemeralSt
orageRequest":"5Gi"}}'
  zen-s3-kopia: '{"jobSettings":{"ttl":300},"podResources":
{"cpuRequest":"500m","cpuLimit":"4","memoryRequest":"500Mi","memoryLimit":"4Gi"}, "storage":{"storageClassName":"ocs-
external-storagecluster-ceph-rbd","size":"25Gi"}}'
```

It ensures that the maintenance jobs related to data from the zen namespace connected to BSL S3 use a 25Gi PersistentVolumeClaim cache and do not set resource requests or limits on ephemeral storage.

node-agent-config

To configure the data mover pods to use PVCs as cache in place of ephemeral storage, add the following field into the node-agent-config config map:

```
storage":{"storageClassName":"ocs-external-storagecluster-ceph-rbd","size":"25Gi"}
```

To configure the data mover pods to use PVCs as cache in place of ephemeral storage.
Example value of the node-agent-config:

```
{"loadConcurrency": {"globalConfig": 5}, "podResources": {"cpuRequest": "2", "cpuLimit": "4", "memoryRequest": "4Gi",
"memoryLimit": "16Gi", "ephemeralStorageRequest": "5Gi", "ephemeralStorageLimit": "5Gi"}, "storage":
{"storageClassName":"ocs-external-storagecluster-ceph-rbd","size":"25Gi"}, "backupPVC": {"ocs-storagecluster-cephfs":
{"storageClass": "ocs-storagecluster-cephfs", "readOnly": true, "spcNoRelabeling": true}}}
```

Latest permissible start time for the backup process

You can set the latest permissible start time for a scheduled backup. This setting is useful whenever multiple backups use the same policy or schedule and the agent gets overloaded frequently with jobs as they all start at the same time. Without the windowEndTime setting, jobs have, by default, up to one hour for the agent to start processing before they are considered hung and subsequently cancelled. You can manually spawn additional agent replicas to handle the spike in job activity and process the backlog of jobs sooner, but these replicas remain idle for the remainder of the day. With the windowEndTime option, the job can remain queued up for processing by the agent for a custom period of time before getting cancelled. This allows for a single agent replica to be fully active for a longer period of time throughout the day without the need to manually create multiple policies or schedules.

Set the windowEndTime field in the **`BackupPolicy`** CR in the 24 hour "HH:MM" format.

Example:

```
spec:
  backupStorageLocation:  kn-aws
  provider: isf-backup-restore
  retention:
    number: 5
    unit: days
  schedule:
    cron: `45 14 * * * '
    timezone: America/Los_Angeles
    windowEndTime: `17:45'
```

In this example, the schedule is set to run daily at 14:45 (2.45 PM). By specifying a windowEndTime of 17:45, you can ensure that the backup has until 5:45 PM to start. If the backup does not start by 5.45 PM due to the load on the agent, it gets cancelled. If the agent does not pick it up and the windowEndTime setting does not exist, then backup gets cancelled after one hour.
Note: For all new **`BackupPolicy`** CRs that have a specific start time (not just a repeat interval), if the windowEndTime is not explicitly specified, a default end time gets set to 4 hours after the scheduled start time. Existing **`BackupPolicy`** CRs maintain their original behavior without windowEndTime, resulting in a one-hour limit before they are considered hung and subsequently cancelled. You can edit the existing Policy CRs to add the windowEndTime option.

Failed backup retention

By default, failed backups are automatically deleted after 7 days. You can change this retention period by updating the **`failedBackupRetention`** configuration parameter in the **`guardian-configmap`** configmap. To do so, add the following entry to the configmap: **`failedBackupRetention: '5'`**. This sets the retention period to 5 days.

FIPS-140-2

The Federal Information Processing Standard Publication 140-2 (FIPS-140-2) is a standard that defines a set of security requirements for the use of cryptographic modules. Law mandates this standard for the US government agencies and contractors and is also referenced in other international and industry specific standards.

IBM Fusion Backup & Restore service is FIPS compliant. It uses FIPS validated cryptographic modules provided by Red Hat Enterprise Linux OS/CoreOS (RHCOS).

Backup & Restore service installation and upgrade issues

Use this troubleshooting information to resolve install and upgrade problems that are related to Backup & Restore service.

One or more Backup & Restore pods are in CrashLoopBackOff state due to startup/liveness/readiness probe failures

Problem statement
: During installation or upgrade of IBM Fusion Backup & Restore service, a few pods might be in **`CrashLoopBackOff`** state due to startup/liveness/readiness probe failures. This causes the installation or upgrade to not progress further.

Cause
: One or more Backup & Restore service pods enter a **`CrashLoopBackOff`** state due to startup/liveness/readiness probe failures. This can happen when the cluster load causes the pod CPU to be throttled leading to longer startup times.

Resolution
: Increase the **`periodSeconds`** and **`failureThreshold`** values for the startup/liveness/readiness probes in the Backup & Restore service deployment. Updating these values allows additional time for the pods to start successfully and enables the service to complete the upgrade.

Backup & Restore service installation may hang at 75%

If the provided custom install namespace exceeds 24 characters in length, then the Fusion Backup & Restore service installation may hang at 75% completion. To resolve this issue, provide a custom namespace that is no more than 24 characters in length.

Application information not repopulated after service reinstallation

Problem statement

The application information gets removed during Backup & Restore service uninstallation. It does not get repopulated in MongoDB after the service reinstallation. Restore jobs validation fail with the following error message.

```
postJobRequest: Application Info not returned from application service for applicationId
```

Resolution

The workaround is to restart the **`isf-application-operator-controller-manager`** pod in the IBM Fusion namespace.

Backup & Restore server installation displays an Invalid Input error

Problem statement

The Backup & Restore service deployment fails with the following error in the IBM Fusion user interface:

`Install Error: Invalid Input: Both the api Server and bootstrapToken fields need to be populated`

Resolution

As a resolution, try with a private browser window.

Backup & Restore hub service in a custom namespace

Problem statement

In general, the Backup & Restore service is installed or upgraded to the default **`ibm-backup-restore`** namespace. To avoid issues during the installation or upgrade of the service with custom namespace, do the resolution steps.

Resolution

1. Go to Operators > Installed Operators > Fusion Service Definition.
2. Open the **`ibm-backup-restore-service`** CR and go to the YAML tab.
3. In spec.onboarding.parameters, search for parameter with **name** as namespace and change the defaultValue to the custom-namespace where the service must be installed.
   Example:

```
parameters:
      - dataType: string
        name: namespace
        defaultValue: <custom-namespace>
        userInterface: false
        required: true
        descriptionCode: BMYSRV00003
        displayNameCode: BMYSRV00004
```

4. Store the following YAML in server-fsi.yaml, and replace **`<custom-namespace>`** with the namespace where the service must be installed.

```
apiVersion: service.isf.ibm.com/v1
kind: FusionServiceInstance
metadata:
  name: ibm-backup-restore-service-instance
  namespace: ibm-spectrum-fusion-ns
spec:
  parameters:
    - name: doInstall
      provided: false
      value: 'true'
    - name: namespace
      provided: false
      value: <custom-namespace>
    - name: storageClass
      provided: true
      value: lvms-lmvg
  serviceDefinition: ibm-backup-restore-service
  triggerUpdate: false
  enabled: true
  doInstall: true
```

5. Run the following command to apply the changes:

```
oc apply -f server-fsi.yaml
```

6. For the upgrade procedure, upgrade the IBM Fusion operator, repeat step 1 of the resolution, and then upgrade the service.

Pods in Crashloopbackoff state after upgrade

Problem statement

The Backup & Restore service health changes to unknown and two pods go into Crashlookbackoff state.

Resolution

In the resource settings of **`guardian-dp-operator`** pod that is in **`ibm-backup-restore`** namespace, set the value of IBM Fusion operator memory limits to 1000 mi.

Example:

```
resources:
          limits:
            cpu: 1000m
            memory: 1000Mi
          requests:
            cpu: 500m
            memory: 250Mi
```

Backup & Restore service goes into unknown state

Problem statement
: This Backup & Restore service might go into unknown state when you upgrade IBM Fusion from 2.6.x to 2.13.0.

Resolution
: It automatically shows healthy on the IBM Fusion user interface after you upgrade the Backup & Restore service.

MongoDB pod crashes with `CrashLoopBackOff` status

Problem statement
: The MongoDB pod crashes due to an OOM error.

Resolution
: To resolve the error, increase the memory limit from 256Mi to 512Mi. Do the following steps to change the memory limit:

1. Log in to the Red Hat® OpenShift® web console as an administrator.
2. Go to Workloads > StatefulSet.
3. Select the project **`ibm-backup-restore`**.
4. Select the MongoDB pod, and go to the YAML tab.
5. In the YAML, change the memory limit for MongoDB container from 256Mi to 512Mi.
6. Click Save.

Backup & Restore service installation and upgrade issue with the IBM Storage Protect Plus catalog source

Problem statement
: The backups can fail with a **`FailedValidation`** error after you install or upgrade the IBM Fusion and Backup & Restore service to 2.8.0.

```
validationErrors': ['an existing backup storage location wasn't specified at backup creation time and the default guardian-min

Error: BackupStorageLocation.velero.io "guardian-minio" not found'
```

: It occurs when the IBM Storage Protect Plus catalog exists in the OpenShift Container Platform clusters.

Resolution
: The workaround is to upgrade the OADP, installed in the **`ibm-backup-restore`** namespace, to version 1.3.

Troubleshooting Backup & Restore service issues

List of known Backup & Restore issues in IBM Fusion.

Small IBM Scale PersistentVolumeClaims (PVCs) can cause failures

Small IBM Scale PersistentVolumeClaims (PVCs) are more prone to VolumeSnapshot failures. The smaller the PVC, the higher the likelihood. However, failures can occur regardless of size. According to IBM Scale, 10Gi is the threshold above which failures are quite unlikely, and below which they may begin to occur on a semi-regular basis.

Common issues

This section outlines common issues and their resolutions across multiple subcategories, such as backup, restore, service protection.

Backup or restore fails because of PVCs

Problem statement
: Backup & Restore fails with the following error message:

```
error to create persistentvolumeclaims "<claim name>" is forbidden: exceeded quota"
```

Cause
: During backup or restore operations, the Backup & Restore service temporarily allocates additional storage through PersistentVolumeClaims to stage application data. The requested storage size can even match the application size backed up or restored.

Resolution

- **Scenario 1: Failure due to ResourceQuota**:
  Adjust or remove the storage.capacity field from the **`ResourceQuota`** object to allow the Backup & Restore service to allocate staging PersistentVolumeClaims as needed. The quota must equal the total size of the largest application's PersistentVolumeClaims, plus an additional 10Gi

for local in-place policy storage. A higher quota may be required when you perform parallel operations that span all applications backed up or restored simultaneously.

- **Scenario 2: Failure due to ClusterResourceQuota**:
  During backup or restore operations, Backup & Restore service temporarily consumes additional storage using PersistentVolumeClaims to stage data. To ensure successful backup and restore, either exclude the Backup & Restore namespace from the quota or increase it sufficiently. The quota must be set to the total size of the largest application's PersistentVolumeClaims, plus an additional 10Gi for local in-place policy storage. A higher quota may be necessary when you perform parallel operations that span all applications backed up or restored simultaneously.

For more information about resource quotas, see Red Hat Documentation.

- **Backup issues**
  List of backup issues in the Backup & Restore service of IBM Fusion.
- **Restore issues**
  List of restore issues in Backup & Restore service of IBM Fusion.
- **IBM Software Hub backup and restore issues**
  This section lists the troubleshooting tips and tricks during backup and restore of IBM Software Hub.
- **Hub and spoke connection issues**
  Procedure to debug issue in the hub and spoke connections. Backup & Restore service uses connection CR to setup hub and spoke connection.
- **Backup & restore service from OpenShift Container Platform console**
  List of issues in the Backup & Restore managed and monitored from the OpenShift® Container Platform console.

Backup issues

List of backup issues in the Backup & Restore service of IBM Fusion.

VolumeSnapShotClass Creation issue

Problem statement

Backup & Restore fails for applications on the remote cluster that is connected through the Global Data Platform Remote Mount Service. The reason is that the `VolumeSnapShotClass` for scale is not created through code as expected.

Diagnosis

Run the following command to confirm whether the `VolumeSnapShotClass` for scale exists or not:

```
oc get VolumeSnapshotClass ibm-spectrum-scale-snapshot-class
```

Example output:

```
NAME                                DRIVER             DELETIONPOLICY    AGE
ibm-spectrum-scale-snapshot-class   spectrumscale.csi.ibm.com       Delete      119d
```

If it does not exist, then do the resolution steps.

Resolution

Do the following steps to resolve the issue:

1. Run the following command to get the `clusterrolebinding` for the CNS operator:

   ```
   oc get clusterrolebindings -o json | jq -r '.items[] | select(.subjects[]?.name == "isf-cns-operator-controller-manager" and .roleRef.name != "system:auth-delegator") | .metadata.name'
   ```

   Example output:

   ```
   % oc get clusterrolebindings -o json | jq -r '.items[] | select(.subjects[]?.name == "isf-cns-operator-controller-manager

   isf-operator.v2.10.0-375-awgs31UnUIdSsrwvNPiqlHYLqL6givWque6aPd
   ```

2. Find the Role name corresponding to the CNS Operator.
3. Run the following command to retrieve information about ClusterRoles in the cluster:
   Example command:

   Replace the provided example values in this command with your own.

   ```
   oc get ClusterRole isf-operator.v2.9.0-87828235c84 -o yaml > clusterrole.yaml
   ```

4. Open the clusterrole.yam in edit mode.

   ```
   vim clusterrole.yaml
   ```

5. Add the following rules in the clusterrole.yaml created in previous step.

   ```
   - apiGroups:
     - snapshot.storage.k8s.io
     resources:
     - volumesnapshotclasses
     verbs:
     - create
     - delete
     - get
     - list
     - patch
     - update
     - watch
   ```

6. Apply the YAML file:

   ```
   oc apply -f clusterrole.yaml
   ```

7. In OpenShift® Container Platform console, go to Storage > VolumeSnapShotClass, and verify whether a class with the name **ibm-spectrum-scale-snapshot-class** exists.

Backup size for block volumes is not correct

When the policy backup storage location gets changed, the backup size for application backups with Ceph RBD block volumes is incorrectly displayed during the first backup. As the correct information is displayed after the first backup following the change in the backup storage location, you can ignore this issue.

Multiple VM backups failing with snapshot deadline exceeded error

Resolution

1. Run the following command to enable full auditing:

   ```
   sudo auditctl -w /etc/shadow -p
   ```

2. Rerun the backup and run the following command to identify the file that is causing ownership or permission issues.

   ```
   sudo ausearch -m avc -ts recent
   ```

3. If the **ausearch** command reports issues, then run the following commands to generate a local policy to allow access.

   ```
   sudo ausearch -c 'qemu-ga' --raw | audit2allow -M my-qemuga
   sudo semodule -X 300 -i my-qemuga.pp
   ```

Failed to create snapshot content

Problem statement

Failed to create snapshot content with the following error:
`Cannot find CSI PersistentVolumeSource for directory-based static volume`

Resolution

To resolve the error, see Create a VolumeSnapshot.

Assign a backup policy operation fails

Problem statement

If you have a PolicyAssignment for an application on the hub and you create a PolicyAssignment for the same application on the spoke, then your attempt to assign a backup policy for the application fails. In both assignments, the application, backup policy, and short-form cluster name are the same. The current format of the PolicyAssignment CR name is **appName-backupPolicyName-shortFormClusterName**. The issue happens when the first string of the cluster names is identical. In this scenario, the creation gets rejected because the PolicyAssignment name exists in OpenShift Container Platform.
For example:

Hub assignment creates **app1-bp1-apps**:

- Application - **app1**
- BackupPolicy - **bp1**
- AppCluster - **apps.cluster1**

Spoke assignment creates **app1-bp1-apps** (The OpenShift Container Platform rejects it)

- Application - **app1**
- BackupPolicy - **bp1**
- AppCluster - **apps.cluster2**

Resolution

To create the PolicyAssignment for the spoke application, delete the PolicyAssignment CR for the hub application assignment and attempt spoke application assignment again.

Backups do not work as defined in the backup policies

Problem statement

Sometimes, backups do not work as defined in the backup policies, especially when you set hourly policies. For example, if you set a policy for two hours and it does not run every two hours, then gaps exist in the backup history. The possible reason might be that during pod crash and restart, scheduled jobs were not accounting for the time zone, causing gaps in run intervals.

Diagnosis

The following are the observed symptoms:

- Policies with custom every X hour at minute YY schedules: the first scheduled run of this policy will run at minute YY after X hours + time zone offset from UTC instead of at minute YY after X hours.
- Monthly and yearly policies run more frequently.

Resolution

You can start backups manually until the next scheduled time.

Backup jobs are stuck in a running state for a long time and are not canceled

Resolution

Do the following steps to resolve the issue:

1. Ensure that all jobs are finished and the queue is empty before you do any disruptive actions like node restarts.
2. If jobs are running for a long period and do not progress, follow the steps to delete the backup or restore CR directly.
   a. Log in to IBM Fusion.
   b. Go to Backup & Restore > Jobs > Queue and get the name of the job that is stuck.
   c. Run the following command to delete backup job.

   ```
   oc delete fbackup <job_name>
   ```

   d. Run the following command to delete restore job.

   ```
   oc delete frestore <job_name>
   ```

Policy creation

Problem statement

Sometimes, when you create a backup policy, the following errors can occur:

```
Error: Policy daily-snapshot could not created.
```

Resolution

Restart the `isf-data-protection-operator-controller-manager-* pod` in IBM Fusion namespace. It triggers the recreation of the in-place-snapshot BackupStorageLocation CR.

Policy assignment from Backup & Restore service page of the OpenShift Container Platform console

Problem statement

In the Backup & Restore service page of the OpenShift Container Platform console, the backup policy assignment to an application fails with a gateway timeout error.

Resolution

Use your IBM Fusion user interface.

Backup of multiple VMs attempt is failed

Problem statement

This issue occurs when some VMs are in a migrating state. The OpenShift Container Platform does not support snapshot of the VMs in migrating state.

Resolution

Follow the steps to resolve this issue:

1. Check whether the virtual machine is in a migrating state:
2. Run the following command to check migrating VM.

   ```
   oc get virtualmachineinstancemigrations -A
   ```

   Example output:

   ```
   NAMESPACE            NAME                                          PHASE         VMI
   fb-bm1-fs-1-5g-10    rhel8-lesser-wildcat-migration-8fhbo          Failed        rhel8-lesser-wildcat
   vm-centipede-bm2     centos-stream9-chilly-hawk-migration-57jyk    Failed        centos-stream9-chilly-hawk
   vm-centos9-bm1-1     centos-stream9-instant-toad-migration-bfyz6   Failed        centos-stream9-instant-toad
   vm-centos9-bm1-1     centos-stream9-instant-toad-migration-d9547   Failed        centos-stream9-instant-toad
   vm-windows10-bm2-1   kubevirt-workload-update-4dm57                Failed        win10-zealous-unicorn
   vm-windows10-bm2-1   kubevirt-workload-update-f2s5w                Failed        win10-zealous-unicorn
   vm-windows10-bm2-1   kubevirt-workload-update-gt6nj                Failed        win10-zealous-unicorn
   vm-windows10-bm2-1   kubevirt-workload-update-rjwmn                Failed        win10-zealous-unicorn
   vm-windows10-bm2-1   kubevirt-workload-update-vfxfl                TargetReady   win10-zealous-unicorn
   vm-windows10-bm2-1   kubevirt-workload-update-z2thw                Failed        win10-zealous-unicorn
   vm-windows11-bm2-1   kubevirt-workload-update-9gr6v                Failed        win11-graceful-coyote
   vm-windows11-bm2-1   kubevirt-workload-update-clbck                Failed        win11-graceful-coyote
   vm-windows11-bm2-1   kubevirt-workload-update-j6pmx                Failed        win11-graceful-coyote
   vm-windows11-bm2-1   kubevirt-workload-update-sfbbx                Pending       win11-graceful-coyote
   vm-windows11-bm2-1   kubevirt-workload-update-th5dd                Failed        win11-graceful-coyote
   vm-windows11-bm2-1   kubevirt-workload-update-zl679                Failed        win11-graceful-coyote
   vm-windows11-bm2-2   kubevirt-workload-update-7dp6g                Failed        win11-conservative-moth
   vm-windows11-bm2-2   kubevirt-workload-update-9nb9m                TargetReady   win11-conservative-moth
   vm-windows11-bm2-2   kubevirt-workload-update-cdrf5                Failed        win11-conservative-moth
   vm-windows11-bm2-2   kubevirt-workload-update-dm8fz                Failed        win11-conservative-moth
   vm-windows11-bm2-2   kubevirt-workload-update-kwr6c                Failed        win11-conservative-moth
   vm-windows11-bm2-2   kubevirt-workload-update-zt8wx                Failed        win11-conservative-moth
   ```

3. Exclude the migrating virtual machine from the backup. Reattempt it after the migration is complete.

Backup applications table does not show the new backup times for the backed-up applications

Problem statement

The backup applications table does not show the new backup times for the backed-up applications.

Resolution
Go to the Applications and Jobs view to see the last successful backup job for a given application. For applications on the hub, the Applications table has the correct last backup time.

Backups are failing for the virtual machines

Problem statement
The backups and snapshots are failing for the virtual machines that is mounted with second disk.

Resolution

1. Run the following command to get disks details for the virtual machine.

```
oc get virtualmachine -A -o json | jq '.items[] | [{name:.metadata.name, namespace:.metadata.namespace,
volumes:.spec.template.spec.volumes}] | select(.[].volumes[].dataVolume | length > 1) | {name
:.[].name, namespace:.[].namespace}'
```

   Example output:

```
{
"name": "rhel9-absent-basilisk",
"namespace": "vmtesting"
}
```

2. If you find the virtual machines are mounted with second disk, then follow the steps mentioned in the Red Hat solution to resolve the issue.

Known issues and limitations

- When the Backup & Restore Hub auto-upgrade setting is disabled, the Backup & Restore service may still be upgraded to IBM Fusion 2.13.0.
- Note: This issue only occurs with the first job after installation.
  After a fresh installation, the first job might not process as expected. To resolve this issue, cancel the job and wait for the cancellation to complete. Then, retry the job to process it successfully.
- The OpenShift Container Platform cluster can have problems and become unusable. After you recover the cluster, rejoin the connections. OpenShift Container Platform cluster can have problems and become unusable.
- The S3 bucket must not have an expiration policy or an archive rule. For more information about this known issue, see S3 buckets must not enable expiration policies.
- The Azure Endpoint URL must not contain the name of the bucket.
- When you backup an application present in namespace with high security context constraints privileges. The restored namespace will not have the same security context constraints privileges, resulting in restored pods in crashloopbackoff status.
  Workaround:
  - Restart the application pod.
- When you configure node affinity for the Backup & Restore, consider the following limitations:
  - There is no way to add `nodeAffinity` in `DataProtectionApplication` CR. The Velero pod may get scheduled on the GPU nodes, even if other nodes are eligible for scheduling.
  - There is one `Node-agent` pod per node, which continues to run on GPU nodes as well.
    For more information about Configuring the OpenShift API for Data Protection with Multicloud Object Gateway, see Red Hat documentation.
  - There is no way to specify `nodeAffinity` for AMQ and OAPD operator pods. These operator pods may get scheduled on the GPU nodes, even if other nodes are eligible for scheduling.
  - There is no way to specify anti `NodeAffinity` for OLM jobs pods, so these pods may get scheduled on the GPU nodes, even if other nodes are eligible for scheduling. However, these are very short lived pods.
  - The `datamover` pod for restore work may get scheduled on a GPU node. Currently, Kopia honors the setting for backup job pod, but not for restore job pod.
  - In summary, the following pods do not have the logic to avoid GPU nodes:
    - Velero
    - Datamover (for restore)
    - AMQ Operator
    - OAPD operator
    - Node Agent
    - OLM jobs

Restore issues

List of restore issues in Backup & Restore service of IBM Fusion.

Restore Fails with `Timeout on Preparing Data Download` error

Problem statement
The restore operation is inconsistent and sometimes fails with the error timeout on preparing data download in the data mover pod due to reason: failed to add finalizer. This issue is observed in certain environments, and the cause is not immediately apparent.

Symptoms

- Restore operation fails with a timeout error.
- Error message indicates a problem with preparing data download in the data mover pod.
- The specific error message is failed to add finalizer.

Cause

The cause of this issue is not definitively known but is related to the functioning of the IBM Storage Scale Container Storage Interface (CSI) pods. The pods may be experiencing issues that prevent the successful completion of the restore operation.

Resolution

To resolve this issue, follow these steps:

1. Run the following command to delete the IBM Storage Scale Container Storage Interface (CSI) pods. For versions of IBM Storage Scale Container Native Storage Access (CNSA) prior to 6.0.1, use **ibm-spectrum-scale-csi** as the namespace instead of **ibm-spectrum-scale**.

   ```
   kubectl delete pod -l product=ibm-spectrum-scale-csi -n ibm-spectrum-scale
   ```

   Example output:

   ```
   pod "ibm-spectrum-scale-csi-attacher-86c4864c94-tlcqn" deleted
   pod "ibm-spectrum-scale-csi-attacher-86c4864c94-wn6sr" deleted
   pod "ibm-spectrum-scale-csi-bqtwj" deleted
   pod "ibm-spectrum-scale-csi-dqsgr" deleted
   pod "ibm-spectrum-scale-csi-mskc6" deleted
   pod "ibm-spectrum-scale-csi-provisioner-67b878b57-6lftl" deleted
   pod "ibm-spectrum-scale-csi-resizer-84cf579d49-bx4jr" deleted
   pod "ibm-spectrum-scale-csi-rrggd" deleted
   pod "ibm-spectrum-scale-csi-snapshotter-8b69bc4dd-xkt4s" deleted
   pod "ibm-spectrum-scale-csi-srh2z" deleted
   ```

2. Run the following command to verify that the IBM Storage Scale Container Storage Interface (CSI) pods restarted successfully.

   ```
   oc get pods -l product=ibm-spectrum-scale-csi -n ibm-spectrum-scale
   ```

   Example output:

   ```
   NAME                                                  READY   STATUS    RESTARTS   AGE
   ibm-spectrum-scale-csi-attacher-86c4864c94-55z9p      1/1     Running   0          7s
   ibm-spectrum-scale-csi-attacher-86c4864c94-t9sww      1/1     Running   0          7s
   ibm-spectrum-scale-csi-frmtq                          3/3     Running   0          6s
   ibm-spectrum-scale-csi-g4mqr                          3/3     Running   0          6s
   ibm-spectrum-scale-csi-hxb4q                          3/3     Running   0          6s
   ibm-spectrum-scale-csi-provisioner-67b878b57-ltj4g    1/1     Running   0          7s
   ibm-spectrum-scale-csi-q5jgp                          3/3     Running   0          6s
   ibm-spectrum-scale-csi-resizer-84cf579d49-wvgv2       1/1     Running   0          7s
   ibm-spectrum-scale-csi-rgv5d                          3/3     Running   0          7s
   ibm-spectrum-scale-csi-snapshotter-8b69bc4dd-7cpq5    1/1     Running   0          7s
   ```

Restore fails because of ephemeral local storage exceeding limits error in Velero pod

Resolution

When you back up and restore a large number of cluster resources, Velero pod resource limits must be increased for the operations to succeed. The optimal values depend on your individual environment, but the following values are representative for backup and restore of 1500 resources. To change the local storage the Velero Pod is allowed to use from the default 500Mi (Mebibytes) to 1Gi (Gibibytes):

```
oc patch dpa velero -n <backup-restore-namespace> --type merge -p '{"spec": {"configuration": {"velero": {"podConfig":
{"resourceAllocations": {"limits": {"ephemeral-storage": "1Gi"}}}}}}}'
```

To change the memory limit of the Velero Pod from default 2Gi (Gibibytes) to 4Gi (Gibibytes):

```
oc patch dpa velero -n <backup-restore-namespace> --type merge -p '{"spec": {"configuration": {"velero": {"podConfig":
{"resourceAllocations": {"limits": {"memory": "4Gi"}}}}}}}'
```

Restore of applications from hub or spoke 2.9.0 to spoke at version 2.8.1 fails

Problem statement

Restore of applications from hub or spoke 2.9.0 to spoke at version 2.8.x fails with error in Transaction Manager service:

```
Argument of type 'NoneType' is not iterable.
```

Resolution

This is a known limitation so do not attempt restore 2.9.0 (kopia) based backups to a lower version 2.8.x (restic).

IBM Cloud Limitations with OADP DataMover

Problem statement

Backups of raw volumes fail for IBM Cloud with OADP 1.4.0 or lower.

Cause

The OADP 1.3 or higher expose the volume of the underlying host during backup or restore. The folders on the host are exposed in the Pods that are associated with the Daemonset node-agent. The /var/lib/kubelet/{plugins,pods} folder is exposed by default. The folders required to work on IBM Cloud are /var/data/kubelet/{plugins,pods}. As a result, the backup and restore of volumeMode: block volumes fail with the following example error:

```
Failed transferring data

[BMYBR0009](https://ibm.com/docs/SSFETU_2.9/errorcodes/BMYBR0009.html) There was an error when processing the job in the Trans
```

The ID of the individual DataUpload varies for jobs.

Resolution

1. Set the DataMover type to **legacy** by either the global method or the Per PolicyAssignment method. For the procedure to update the type, see Configure DataMover type for Backup and Restore.
   Though this workaround allows continued use of of DataMover **kopia** type, it has the following drawbacks:

It disables further changes to DataMover and Velero configurations, such as the ability to change resource allocations (CPU, Memory, and Ephemeral-storage) and nodeSelectors (datamover node placement) for Datamover type **kopia**.

Legacy is not affected by this workaround.

To avoid job failures, do not make these changes while a backup or restore job is in progress.

2. In the OpenShift Console, go to Workloads > Deployments.
3. Select **openshift-adp-controller-manager** and scale the number of Pods to 0.
4. Go to Workloads > Daemonsets and select node-agent.
5. Select the YAML tab.
6. Under the volumes section, add the additional volume **host-data** as shown in the following example.
   Note: It exposes an additional folder on the host other than the folders mention in the Cause of this issue.

```
    volumes:
       - name: host-pods
         hostPath:
           path: /var/lib/kubelet/pods
           type: ''
       - name: host-plugins
         hostPath:
           path: /var/lib/kubelet/plugins
           type: ''
       - name: host-data
         hostPath:
           path: /var/data/kubelet
           type: ''
       - name: scratch
         emptyDir: {}
       - name: certs
         emptyDir: {}
```

7. Under **volumeMounts**, add the **host-data** volume as shown in the following example.

```
          volumeMounts:
            - name: host-pods
              mountPath: /host_pods
              mountPropagation: HostToContainer
            - name: host-plugins
              mountPath: /var/lib/kubelet/plugins
              mountPropagation: HostToContainer
            - name: host-data
              mountPath: /var/data/kubelet
              mountPropagation: HostToContainer
            - name: scratch
              mountPath: /scratch
            - name: certs
              mountPath: /etc/ssl/certs
```

8. Save the changes and wait for a couple minutes for the Pods to restart.
   The backups and restores of PersistentVolumeClaims with volumeMode: block succeeds on Red Hat® OpenShift® of IBM Cloud.

Restore fails on IBM Power Systems

Restore failures are observed in a IBM Power Systems environment. If you have cluster(s) running on IBM Power Systems, do not upgrade the Backup & Restore service to 2.8.1 instead contact Contact IBM Support.

exec format error

Problem statement

Sometimes, you may observe the following error message:

```
"exec <executable name>": exec format error
```

For example:

```
The pod log is empty except for this message: exec /filebrowser
```

The example error can be due to the wrong architecture of the container. For example, an amd64 container on s390x nodes or an s90x container on amd64 nodes.

Resolution

As a resolution, check whether the container that you want to restore and the local node architecture match.

Restore of namespaces that contains admission webhooks fails

Problem statement

Restore of namespaces that contains admission webhooks fails.
Example error in IBM Fusion restore job:

```
"Failed restore <some  resource>" "BMYBR0003
      RestorePvcsFailed There was an error when  processing the job in the Transaction Manager
      service"
```

Example error in Velero pod:

```
level=error msg="Namespace
      domino-platform, resource restore error: error restoring
      certificaterequests.cert-manager.io/domino-platform/hephaestus-buildkit-client-85k2v:
      Internal error occurred: failed calling webhook  "webhook.cert-manager.io": failed to call
      webhook: Post  "https://cert-manager-webhook.domino-platform.svc:443/mutate?timeout=10s\\":
      service "cert-manager-webhook" not found"
```

Resolution

1. Identify the admission webhooks that is applicable to the namespace being restored:

   ```
   oc get mutatingwebhookconfigurations
   oc describe mutatingwebhookconfigurations
   ```

2. Change the failure Policy parameter from `Fail` to `Ignore` to temporarily disable webhook validation prior to restore:

   ```
   failurePolicy: Ignore
   ```

Restore before upgrade fails with a BMYBR0003 error

Problem statement

When you try to restore backups before upgrade, it fails with a BMYBR0003 error.

Diagnosis

After you upgrade, your jobs may fail:

- Backup jobs with the status:

  ```
  "Failed transferring data" "BMYBR0003 There was an error when processing the job in the Transaction Manager service"
  ```

- Restore jobs with the status:

  ```
  "Failed restore <some resource>" "BMYBR0003 There was an error when processing the job in the Transaction Manager service
  ```

Confirm the issue in the logs of the manager container of the Data Mover pod.

A sample error message:

```
2023-07-26T03:39:47Z    ERROR   Failed with error.      {"controller": "guardiancopyrestore", "controllerGroup": "guardian.isf
github.ibm.com/ProjectAbell/guardian-dm-operator/controllers/util.Retry
        /workspace/controllers/util/utils.go:39
github.ibm.com/ProjectAbell/guardian-dm-operator/controllers/kafka.(*kafkaWriterConnection).PublishMessage
        /workspace/controllers/kafka/kafka_native_connection.go:71
github.ibm.com/ProjectAbell/guardian-dm-operator/controllers.(*GuardianCopyRestoreReconciler).updateOverallCrStatus
        /workspace/controllers/status.go:191
github.ibm.com/ProjectAbell/guardian-dm-operator/controllers.(*GuardianCopyRestoreReconciler).doRestore
        /workspace/controllers/guardiancopyrestore_controller.go:187
github.ibm.com/ProjectAbell/guardian-dm-operator/controllers.(*GuardianCopyRestoreReconciler).Reconcile
        /workspace/controllers/guardiancopyrestore_controller.go:92
sigs.k8s.io/controller-runtime/pkg/internal/controller.(*Controller).Reconcile
        /go/pkg/mod/sigs.k8s.io/controller-runtime@v0.14.6/pkg/internal/controller/controller.go:122
sigs.k8s.io/controller-runtime/pkg/internal/controller.(*Controller).reconcileHandler
        /go/pkg/mod/sigs.k8s.io/controller-runtime@v0.14.6/pkg/internal/controller/controller.go:323
sigs.k8s.io/controller-runtime/pkg/internal/controller.(*Controller).processNextWorkItem
        /go/pkg/mod/sigs.k8s.io/controller-runtime@v0.14.6/pkg/internal/controller/controller.go:274
sigs.k8s.io/controller-runtime/pkg/internal/controller.(*Controller).Start.func2.2
        /go/pkg/mod/sigs.k8s.io/controller-runtime@v0.14.6/pkg/internal/controller/controller.go:235
```

Resolution:

Search for the `guardian-dm-controller-manager` and kill it. A new pod starts in a minute. After the pod reaches a healthy state, retry backup and restores.

"Failed restore snapshot" error occurs with applications using IBM Storage Scale storage PVCs

- A "Failed restore snapshot" error occurs with applications using IBM Storage Scale storage PVCs.

  Cause

  The "disk quota exceeded" error occurs whenever you restore from an object storage location having applications that use IBM Storage Scale PVC with a size less than 10 GB.

  Resolution

  Increase the IBM Storage Scale PVC size to a minimum of 10 GB and do a backup and restore operation.

Restore to a cluster that does not have an identical storage class as the source cluster

You cannot restore to a cluster that does not have an identical storage class as the source cluster. However, the transaction manager still attempts to create PVCs with the non-existent storage class on the spoke cluster and eventually fails with `Failed restore snapshot` status.

Applications page does not show the details of the application

Problem statement

The new backed up applications page does not show the details of the application when you upgrade IBM Fusion to the latest version while leaving the Backup & Restore service in the older version.

Resolution

As a resolution, it is recommended to upgrade the Backup & Restore service to the latest version after an IBM Fusion upgrade.

S3 buckets must not enable expiration policies

Backup & Restore currently uses restic to move the Velero backups to repositories on S3 buckets. When a restic repository gets initialized, it creates a configuration file and several subdirectories for its snapshots. As restic does not update it after initialization, the modification timestamp of this configuration file is never updated.

If you configure an expiration policy on the container, the restic configuration objects get deleted at the end of the expiration period. If you have an archive rule set, the configuration objects get archived after the time defined in the rule. In either case, the configuration is no longer accessible and subsequent backup and restore operations fail.

Note: The S3 buckets must not enable expiration policies. Also, the bucket must not have an archive rule set.

Virtual machine restore failure

Problem statement

By default, the VirtualMachineClone objects are not restored due to the following unexpected behavior:

- If you create a VirtualMachineClone object and delete the original virtual machine, then the restore fails because the object gets rejected.
- If you create a VirtualMachineClone object and then delete the clone virtual machine, then the restore fails because the Virtualization ignores the **`status.phase`** "Succeeded" and clones the virtual machine again.
  As a result, the clone gets re-created every time you delete it.

- If you create a VirtualMachineClone and then do a backup with the original and clone virtual machine, the restore fails because it ignores the **`status.phase`** "Succeeded" and tries to clone again to the virtual machine that exists.
  The Openshift Virtualization creates a snapshot of the original VirtualMachine, which adds an unwanted VirtualMachineSnasphot and a set of associated VolumeSnapshots after restore whose name starts with "tmp-". The clone operation does not complete and remains stuck in "RestoreInProgress" state because the requested VirtualMachine exists in the VirtualMachineClone.

Resolution

As a resolution, force the restore of the VirtualMachineClone objects by explicitly including it in the Recipe.
Change the OADP **`DataProtectionApplication`** object "velero" and add in the **`spec.configuration.velero.args.restore-resource-priorities`** field as follows:

```
    velero:
      args:
        restore-resource-priorities:
"securitycontextconstraints,customresourcedefinitions,namespaces,managedcluster.cluster.open-cluster-
management.io,managedcluster.clusterview.open-cluster-management.io,klusterletaddonconfig.agent.open-cluster-
management.io,managedclusteraddon.addon.open-cluster-
management.io,storageclasses,volumesnapshotclass.snapshot.storage.k8s.io,volumesnapshotcontents.snapshot.storage.k8s.io,vo
lumesnapshots.snapshot.storage.k8s.io,datauploads.velero.io,persistentvolumes,persistentvolumeclaims,serviceaccounts,secre
ts,configmaps,limitranges,pods,replicasets.apps,clusterclasses.cluster.x-
k8s.io,endpoints,services,-,clusterbootstraps.run.tanzu.vmware.com,clusters.cluster.x-
k8s.io,clusterresourcesets.addons.cluster.x-k8s.io,virtualmachines.kubevirt.io,virtualmachineclones.clone.kubevirt.io"
```

Service recovery involving a large number of CRs

Problem statement

When service recovery processes a large number of CRs, service discovery may fail after you add the backup storage location.

Resolution

1. Check whether **`backup-location`** pod is evicted due to the ephemeral storage limit exceeded.
2. If no **`backup-location`** pod is evicted, check the logs of backup-location pod for any of the following messages:
   - `Processing service protection discovery was successful`
   - `Processing service protection discovery failed`
   - `Error processing service protection discovery`

   If you do not see any of these messages, then the service discovery may be still in running state.

3. If the **`backup-location`** pod restarts because of ephemeral storage limit, update the resources of backup location deployment temporarily. Increase the ephemeral storage limit from 512 Mi to 1 Gi, and the memory limit from 512 Mi to 2 Gi. These limits must be changed back after the service discovery completes populating the service backups that are available for restore.

   ```
   oc set resources deployment backup-location-deployment --limits=ephemeral-storage=1Gi,memory=2Gi -n <backup-restore-
   namespace>
   ```

4. After the service discovery completes, search the **`backup-location`** pod logs for the following message, which includes the number of resources found in the latest, recoverable service backup:

   ```
   Restic backup validation processed
   ```

   Example resources:
   - 363 Application CRs
   - 40 Fusion BackupStorageLocation CRs
   - 385 Secrets for BackupStorageLocation CRs
   - 25 BackupPolicy CRs
   - 565 PolicyAssignment CRs
   - 12126 Backup CRs

   Note: There may be more than one such message if service backups are not suitable for recovery, such as containing no Backup CRs.
   For more information about Backup & Restore Hub performance and scaling, see Backup & restore hub performance and scaling.
5. Update the resources of the transaction manager to temporarily increase the memory limit from 500 Mi to 2 Gi.
   For example:

   ```
   oc set resources deployment transaction-manager --limits=memory=2Gi -n <backup-restore-namespace>
   ```

6. Update the resources of backup services to temporarily increase the number of replicas. For example, 8 for the previous step example, and increasing the memory limit from 1 Gi to 2 Gi.

   ```
   oc set resources deployment backup-service --limits=memory=2Gi -n <backup-restore-namespace>
   ```

   Sample output:

   ```
   The following command isn't working; scale the number of replicas in OpenShift UI.
   oc scale deployment backup-service --replicas=8 -n <backup-restore-namespace>
   ```

7. Update the resources of mongodb based on the number of CRs to be processed. In this example, temporarily increase the CPU limit to 4 and the memory limit to 4 Gi.

   ```
   oc set resources sts mongodb --limits=cpu=4,memory=4Gi -n <backup-restore-namespace>
   ```

8. Update the resources of velero resources. Modify the `DataProtectionApplication` CR named Velero:

   ```
       velero:
         customPlugins:
           - image: 'cp.icr.io/cp/fbr/guardian-kubevirt-velero-
   plugin@sha256:bce6932338d8e2d2ce0dcca2c95bdfa8ab6e337a758e039ee91773f3b26ceb06'
             name: isf-kubevirt
         defaultPlugins:
           - openshift
           - aws
         noDefaultBackupLocation: false
         podConfig:
           labels:
             app.kubernetes.io/part-of: ibm-backup-restore
           resourceAllocations:
             limits:
               cpu: '2'
               ephemeral-storage: 500Mi
               memory: 2Gi
             requests:
               cpu: 200m
               ephemeral-storage: 275Mi
               memory: 256Mi
   ```

   In this example, update the ephemeral-storage limit from 500Mi to 1Gi.
9. Wait for all pods that were modified to restart and be in running state.
   Note: After the service recovers successfuly, it may take time for the services to processes the restored CRs.
10. Wait for all of the backup storage locations to be connected.
11. Check the backup service pods to determine if any CRs are still in progress.
12. After processing completes, revert the resources to their original values or adjust them based on the expected number of concurrent jobs.

More than one BackupRepository found for workload namespace

Problem statement
: When you restore a namespace with multiple PVCs, the restore job might fail with the following error message:

  ```
  error to initialize data path: error to ensure backup repository <repo>: failed to wait BackupRepository, errored early: more
  ```

Cause
: It is due to a race condition during the creation of the Backup repository.

Resolution
: To workaround the problem, clean the restored namespaces and try again.

IBM Software Hub backup and restore issues

This section lists the troubleshooting tips and tricks during backup and restore of IBM Software Hub.

Original namespace as destination for IBM Software Hub application

Problem statement
: OpenShift® restore from IBM Storage Protect Plus does not force you to select the original namespace as destination for IBM Software Hub application.

Impact
: From IBM Storage Protect Plus user interface, IBM Software Hub restore fails whenever you do not select the original namespace as the target destination to restore to.

Resolution
: Whenever you restore IBM Software Hub from IBM Storage Protect Plus user interface, ensure that you select the original namespace as the restore destination. These steps are applicable while you backup and test restore of IBM Software Hub applications. For more information about the procedure, see https://www.ibm.doc/docs/en/SSQNUZ_4.5_test/cpd/admin/online_bar_fusion_spp.html.

Tethered namespace issues

Problem statement
: For IBM Software Hub Application type, you can tether secondary or other namespaces to the primary IBM Software Hub platform namespace. However, after a successful restore, the IBM Fusion application associated with the IBM Software Hub namespace does not restore the tethered namespace.

Resolution

1. Uninstall `cpdbr-oadp` and reinstall it.
2. After you add the tethered namespaces, restart the `isf-application-controller-manager -xxxx` pod that is within the `ibm-spectrum-fusion-ns` to force a reconcile..
3. In about five minutes, run the following command to verify whether the `fapp` is updated to include the tethered namespace in `spec.includedNamespaces`

   ```
   oc get fapp -n ibm-spectrum-fusion-ns cpd-instance -o yaml
   ```

IBM Software Hub backup of zen namespace fails partially in IBM Storage Protect Plus user interface

Problem statement

For an on-demand backup of a single application from a policy that has multiple applications assigned, a snapshot backup is taken only for that selected application but the copy backup is initiated for all of the applications assigned to the policy. This behavior causes the copy backup of the applications without a snapshot backup to fail. This issue does not impact the application selected for the ad-hoc backup.

Backup fails for the platform with error in EDB Postgres cluster

Cause

Labels and annotations in the EDB Postgres cluster resources were not updated after a switchover of the EDB Postgres cluster's primary instance and replica.

Diagnosis

For example, in IBM Fusion, the backup fails at the `Hook: br-service hooks/pre-backup` stage in the backup sequence.
In the `cpdbr-oadp.log` file, you see the following error:

```
time=<timestamp> level=info msg=cmd stderr: Error: cannot take a cold backup of the primary instance
or a target primary instance if the k8s.enterprisedb.io/snapshotAllowColdBackupOnPrimary annotation is not set to enabled
```

Resolution

The following workaround will automatically run before you create a backup. This workaround is useful if you set up automatic backups.
Note: This resolution works only for online backup and restore.

1. Download the edb-patch-aux-ckpt-cm.yaml file.
2. Run the following command:

   ```
   oc apply -n ${PROJECT_CPD_INST_OPERATORS} -f edb-patch-aux-ckpt-cm.yaml
   ```

3. Retry the backup.

Hub and spoke connection issues

Procedure to debug issue in the hub and spoke connections. Backup & Restore service uses connection CR to setup hub and spoke connection.

You might encounter an error when you attempt setup connections between clusters.

Bootstrap token in init secret is not correct or expired: Unauthorized

Problem statement

Connection setup fails with the following message in the connection CR:

```
apiVersion: application.isf.ibm.com/v1
kind: Connection
metadata:
  name: <connection-name>
  namespace: <connection-namespace>
spec:
  remoteCluster:
    apiEndpoint: <cluster api endpoint>
    connectionOperatorNamespace: <connection-namespace>
    heartBeatInterval: 10m
    initSecretName: <init-secret-name>
status:
  conditions:
    - lastTransitionTime: '2023-06-15T02:31:01Z'
      message: 'Bootstrap token in init secret is not correct or expired: Unauthorized'
      reason: CreateBootstrapSecret
      status: 'False'
      type: BootstrapSecretAvaliable
  connectionFromRemoteClusterHealth:
    message: ''
    messageCode: ''
    messageType: ''
  connectionState: Failed
  connectionToRemoteClusterHealth:
    message: ''
    messageCode: ''
    messageType: ''
```

Cause
The bootstrap token in the `init` secret is not correct or expired.

Resolution

1. Get the bootstrap token again.

   ```
   oc create token isf-application-operator-cluster-bootstrap -n <connection-namespace>
   ```

2. Replace the token in `init` secret:

   ```
   oc edit secret <init-secret-name> -n <connection-namespace>
   ```

CA certificate of peer cluster is not correct

Problem statement
The `CA certificate of peer cluster is not correct` error occurs in connection CR.

Cause
The CAcert in the configmap `kube-root-ca.crt` in namespace `kube-public` of the remote cluster is not correct.

Resolution
In the remote cluster, place the right CAcert in the configmap `kube-root-ca.crt` and namespace `kube-public`. Connection pkg also provides a customized configmap.
If it is not possible to place the right CAcert in configmap `kube-root-ca.crt` and namespace `kube-public`, then place the right CAcert in `custom-ca.crt` and Fusion namespace:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: custom-ca.crt
  namespace: <connection-namespace>
data:
  ca.crt: <right CAcert>
```

Backup & restore service from OpenShift Container Platform console

List of issues in the Backup & Restore managed and monitored from the OpenShift® Container Platform console.

fusion-console plugin issue

Problem statement
The OpenShift Container Platform console pod gets into panic state initially with a connection error.

Cause
This error occurs because the inter-namespace communication gets blocked between OpenShift Container Platform console pod (`openshift-console` namespace) and `fusion-console` pod (running in IBM Fusion operator install namespace).

Resolution

1. Check whether any typo exists in the *.apps definition or there is a bad name resolution.
2. If the error persists even after you fix the issue in step 1, restart the Fusion proxy pods.
3. Verify that no `NetworkPolicy/NetworkPolicies` or networking plugin configurations exist that can prevent or interfere with inter-namespace communication.

Known issue

The Fusion Native Console plugin follows the n, n-1 & n-2 support matrix of OpenShift Container Platform, where "n" is an even OpenShift Container Platform version. For example, Fusion Console 2.10 on OpenShift Container Platform 4.18 release. In case of versions that are not supported, run the following command to disable the plugin:

```
oc patch -n ibm-spectrum-fusion-ns ISFConsolePlugin isf-console --type=merge -p '{"spec":{"enabled": 'false'}}'
```

Here, the `ibm-spectrum-fusion-ns` is the namespace where the IBM Fusion operator is installed.

Data Cataloging

Companies need the ability to use unstructured data to meet their business priorities.

IBM Data Cataloging is a container native modern metadata management software that provides data insight for exabyte-scale heterogeneous file, object, backup, and archive storage on premises and in the cloud. The software connects to these data sources to rapidly ingest, consolidate, and index metadata for billions of files and objects.

IBM Data Cataloging provides a rich metadata layer that enables storage administrators, data stewards, and data scientists to efficiently manage, classify, and gain insights from massive amounts of data. It improves storage economics, helps mitigate risk, and accelerates large-scale analytics to create competitive advantage and speed critical research.

Many companies face significant challenges to manage their data. Some difficult challenges that companies face include:

- Pinpointing and activating relevant data for large-scale analytics.
- Lacking the fine-grained visibility needed to map data to business priorities.
- Removing redundant, trivial, and obsolete data.
- Identifying and classifying sensitive data.

Benefits of IBM Data Cataloging

IBM Data Cataloging can help you manage your unstructured data by reducing the data storage costs, uncovering hidden data value, and reducing the risk of massive data stores. See Table 1.

Table 1. Benefits of IBM Data Cataloging

Optimize - Improve storage usage	Analyze - Uncover hidden data value	Govern - Mitigate risk and improve data quality	Data Management
Decreases storage capital expenditure (CapEx) by facilitating data movement to colder, cheaper storage.	Accelerates data identification for large-scale analytics.	Perform data inspection and classification.	Automate tags for custom insight.
Increases storage efficiency by eliminating trivial or redundant data.	Operationalize tasks to reduce the burden of data preparation.	Helps ensure that data is compliant with governance policies by labeling sensitive data.	Create reports for analysis.
Reduces storage operating expenditure (OpEx) by improving storage administrator productivity.	Orchestrates the ML/DL and Platform Symphony® MapReduce process.	Helps reduce risk that is hidden in heterogeneous data sources.	GUI search for real-time results Search content for fast discovery.

- **IBM Data Cataloging architecture**
  IBM Data Cataloging is an extensible platform that provides exabyte scale data ingest, data visualization, data activation, and business-oriented data mapping.
- **Installing IBM Data Cataloging**
  IBM Data Cataloging service is a modern metadata management software that provides data insight for exabyte-scale heterogeneous file, object, backup, and archive storage on premises and in the cloud. It can help you manage your unstructured data by reducing the data storage costs, uncovering hidden data value, and reducing the risk of massive data stores.
- **Uninstalling IBM Data Cataloging**
  Steps to uninstall IBM Data Cataloging by using command line interface.
- **Configuring IBM Data Cataloging**
  This section provides information on how to deploy and configure IBM Data Cataloging capabilities.
- **Administering IBM Data Cataloging**
  This section provides information on how to administering the IBM Data Cataloging capabilities.
- **Monitoring IBM Data Cataloging**
  This section provides information on how to monitor the IBM Data Cataloging capabilities.
- **Collecting logs and metrics**
  Steps to collect logs and metrics for the IBM Data Cataloging.
- **Data Cataloging Harvester CLI**
  IBM Data Cataloging Harvester is a new capability that is designed to import external data to the IBM Data Cataloging service catalog database. It imports the data even if it is not coming from a Db2 database that uses the same schema.
- **FKEY migration script**
  The FKEY migration script is a fix to avoid potential rare collisions in the FKEY file identifier when ingesting billions of records.
- **Configurable Db2 log trimmer**
  Db2 log trimmer tool provides a mechanism to trim the informative logs that are generated by scanning a data source. It uses the Harvester CLI to ingest data into the IBM Data Cataloging.
- **IBM Data Cataloging service installation and upgrade issues**
  Use this troubleshooting information to resolve install and upgrade problems that are related to the IBM Data Cataloging service.
- **IBM Fusion IBM Data Cataloging known issues**
  List of all issues in the IBM Data Cataloging service along with its resolution.

IBM Data Cataloging architecture

IBM Data Cataloging is an extensible platform that provides exabyte scale data ingest, data visualization, data activation, and business-oriented data mapping.

Note: All references to "Spectrum Discover" in the images refer to "Data Cataloging".

Exabyte-scale data ingest

- Scan billions of files and objects in a day
- Real-time event notifications
- Automatic indexing

Data Visualization

- Fast queries of billions of records
- Multi-faceted search
- Drilldown Dashboard

Data Activation

- Application software development kit (SDK)
- Extensible architecture

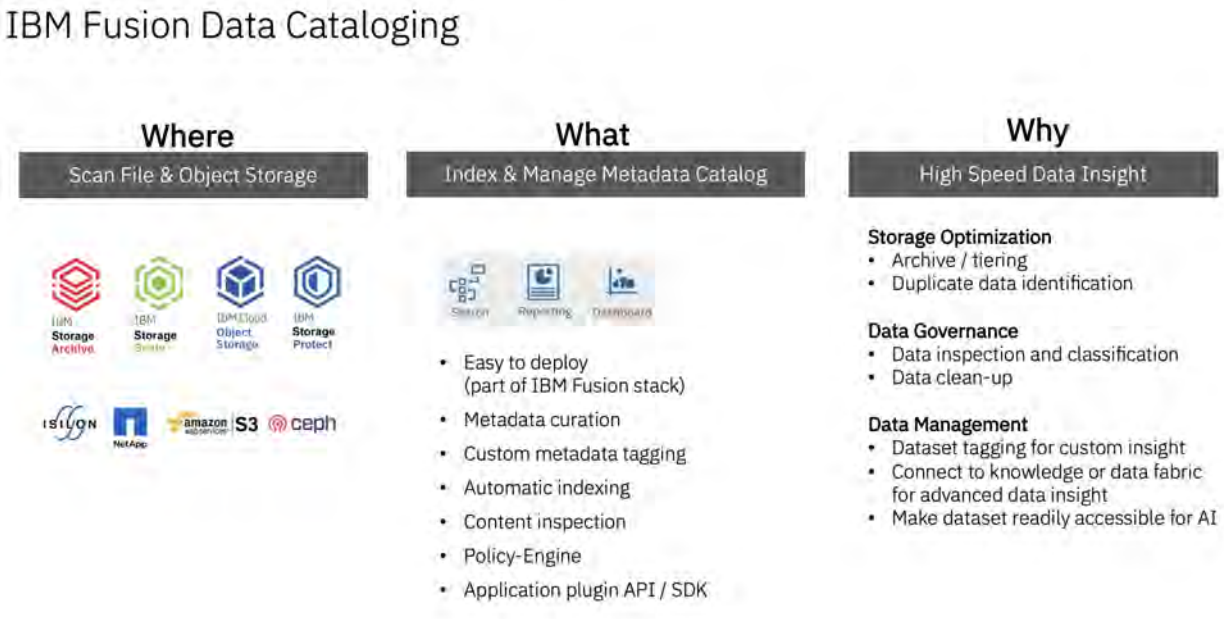
- Solution blueprints

Business-oriented data mapping

- System-level data tagging
- Contextual data tagging
- Policy-driven workflows

The following figure illustrates a high-level view of the IBM Data Cataloging architecture.

Figure 1. IBM Data Cataloging architecture



Data management

IBM Data Cataloging connects to the data sources shown in the architecture image (*IBM Data Cataloging architecture*), and automatically harvests and indexes the system metadata where the system metadata refers to certain information. This might include the following information.

- It might include the names of the files and objects.
- It might include the bucket or path where the data resides.
- It might include the size.
- It might include the time the data sources were last modified.

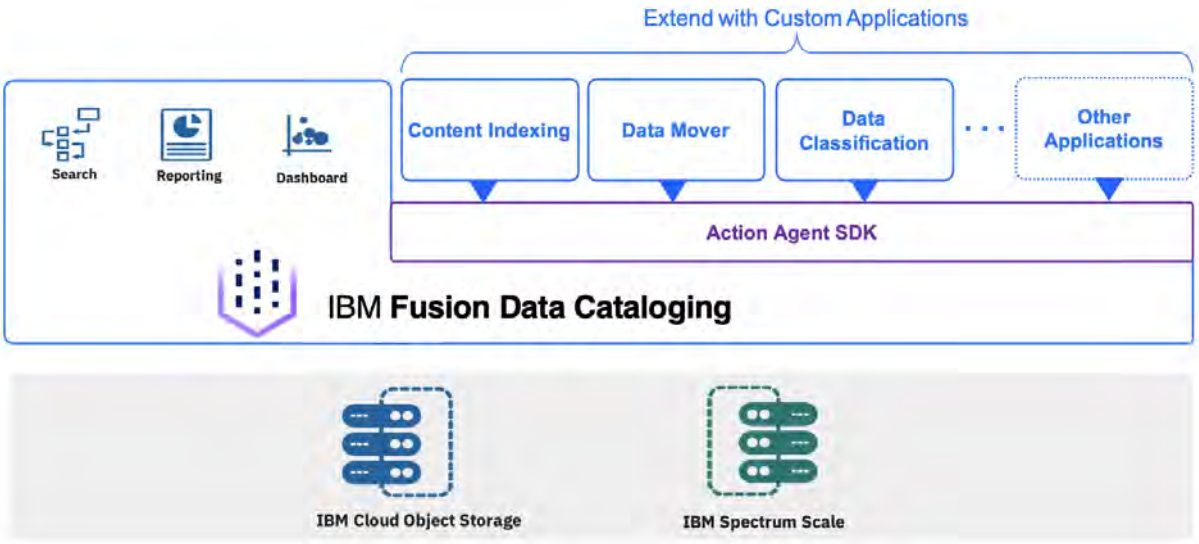
After the data is ingested, analytics are automatically applied to classify and group the data according to the different system metadata attributes. The data can be inspected automatically in IBM Data Cataloging by using built-in content search capabilities to identify sensitive and personally identifiable information and perform data classification. The content inspection capabilities can also be used by researchers and data scientists to extract content from their data sets. This easy-to-use extraction ability assists with data discovery.

The records that are maintained by IBM Data Cataloging can also be further enriched with custom metadata tags that map the data to business constructs and further increase the value of the data.

You can use IBM Data Cataloging catalog to gain insight about your data and to find your data.

The IBM Data Cataloging architecture also supports a community-supported catalog of open source applications that enhance and customize the capabilities of IBM Data Cataloging with third-party extensions. Users can find and install available applications and can develop and share new applications that use an SDK that contains sample code and a fully published API. For more information, see Creating your own applications to use in the Data Cataloging application catalog. For more information, see the topic *Creating your own applications to use in the IBM Data Cataloging application catalog* in the Administration section.

Figure 2. Application SDK architecture



- **Role-based access control**
  IBM Data Cataloging provides access to resources based on roles. You can restrict access to information based on roles.
- **Data source connections**
  A data source connection specifies the parameters for cataloging of metadata from a source system to IBM Data Cataloging.
- **Cataloging metadata**
  System metadata is created and updated by the host system, and not the application software. IBM Data Cataloging allows the addition of tags that can capture non-system metadata-specific attributes, which are stored in the IBM Data Cataloging catalog.
- **Enriching metadata**
  IBM Data Cataloging can enrich the metadata from supported platforms with additional information by using policies, content inspection, custom tags, and custom applications.
- **Graphical user interface**
  The IBM Data Cataloging graphical user interface is a portal that is used for running data searches, report generation, policy and tag management, and user Access Management. Based on a user's role, they might have access to one or more of these areas.
- **Reports for IBM Data Cataloging**
  Reports for IBM Data Cataloging are grouped or non-grouped. Grouped reports have information for count and sum in columns and non-grouped reports have information in rows.

Role-based access control

IBM Data Cataloging provides access to resources based on roles. You can restrict access to information based on roles.

The role that is assigned to a user or group determines the privileges for that user or group. Users and groups can be associated with collections, which use policies that determine the metadata that is available to view.

User and group access can be authenticated by IBM Data Cataloging, an LDAP server, or the IBM Cloud® Object Storage System. The administrator can manage the user access functions.

Roles

Roles determine how users and groups access records or the IBM Data Cataloging environment.
Remember: If a user or group is assigned to multiple roles, the least restrictive role is applicable.
For example, if you are assigned the role of Data User, and you are also assigned the role of a Data Admin, you have the privileges of a Data Admin.

Admin
: An admin can create users, groups, collections, manage LDAP, and IBM Cloud Object Storage connections for user access management.

Data Admin
: Users with the Data Admin role can access all metadata that is collected by IBM Data Cataloging and is not restricted by collections.

Collection Admin
: The Collection Admin role is as a bridge between the Data Admin role and the Data User role.

  - Users with the Collection Admin role can list any type of tag and create or modify `Characteristic` tags. Users with the Collection Admin role cannot create, modify, or delete `Open` and `Restricted` tags. These permissions are the same permissions as the Data User role.
    Note: The built-in `Collection` tag is a special tag that can be set only by users with the Data Admin role. All other tags can be set by any user with the Data User or Data Admin or Collection Admin role.
  - Users with the Collection Admin role can
    - Create, update, and delete the policies for the collections they administer.
    - View, update, and delete policies of data users for the collections they administer. They cannot delete a policy if it has a collection that they do not administer.
    - Add users to collections that they administer. These data users can access a particular collection, which means that they can access the records that are marked with that collection value.

Collection User
: Users with this role can access metadata that is collected by IBM Data Cataloging, but metadata access can be restricted by the collections that are assigned to users in this role.

  - Users assigned with the Collection User role can:
    - Run scans of collections the user is assigned.
    - View policies of the collections the user is assigned.
    - List any type of tag.
  - Users assigned with the Collection User role cannot:
    - Create, update, and delete any policies.
    - Create, modify, and delete any tags.

Data User
: Users with the Data User role can access metadata that is collected by IBM Data Cataloging. Metadata access might be restricted by policies in the collections that are assigned to users in this role. A user with the Data User role can also define tags and policies based on the collections to which the role is assigned.

Service User
: The Service User role is assigned to accounts for IBM® service and support personnel.

Data source connections

A data source connection specifies the parameters for cataloging of metadata from a source system to IBM Data Cataloging.

Without the proper connection information, ingesting metadata from a connected system fails. You can use the Data Source Connections page to view connection information for the data sources that are connected to your environment.

For more information, see Configure data source connections.

Cataloging metadata

System metadata is created and updated by the host system, and not the application software. IBM Data Cataloging allows the addition of tags that can capture non-system metadata-specific attributes, which are stored in the IBM Data Cataloging catalog.

Scans are jobs that are scheduled or on demand, and occur at a data source level. For example, a file system or object vault. A set of metadata records is generated with each record that captures the state of an individual file or object within the data source at the time of the scan.

IBM Data Cataloging supports scanning the following data sources:

- IBM Storage Scale
- IBM Elastic Storage® Server
- IBM Cloud® Object Storage
- IBM Storage Protect
- IBM Spectrum® Archive
- Red Hat® Ceph® Storage
- NetApp Storage Solutions
- Dell EMC Isilon Scale Out Network Attached Storage
- Amazon Simple Storage Service (Amazon S3)
- NFSv3 and NFSv4
- SMB

Live event notifications are triggered by user actions on the source data. Examples are reading, writing, moving, deleting data, changing permissions, or ownership. The events generate a metadata record in real time that is stored in IBM Data Cataloging. IBM Data Cataloging supports live event notifications for the following data sources:

- IBM Cloud Object Storage
- IBM Elastic Storage Server
- IBM Storage Scale
- Red Hat Ceph Storage

With IBM Storage Scale you can enable live events to start a watch folder on the specified file system. The IBM Storage Scale watch folder works with IBM Data Cataloging to capture file system event notifications and deliver them to IBM Data Cataloging by using Kafka.
Important: If the connection from IBM Storage Scale to IBM Data Cataloging is interrupted, the watch suspends. Additionally, events are no longer captured in IBM Data Cataloging, which requires a file system rescan to capture the lost updates.

Enriching metadata

IBM Data Cataloging can enrich the metadata from supported platforms with additional information by using policies, content inspection, custom tags, and custom applications.

Policies

Policies are used to add additional information about the source data that is indexed in IBM Data Cataloging. A policy determines the set of files to add tag values to, or to send to the built-in content inspection capabilities of IBM Data Cataloging, or to a custom application through filtering criteria. The policies give you the ability to run actions one time or on a set schedule. Policies work in batches and can be paused, resumed, stopped, or restarted. You can control the load on the IBM Data Cataloging system or source storage system for content inspection policies and policies that start custom applications.

Applications

A deep inspect application extracts information from source data records and returns it to IBM Data Cataloging to be indexed. For example, by using a custom application, you might create a DEEP-INSPECT policy to extract key characteristics from files of a certain type. The characteristics are applied to the metadata records for the files in IBM Data Cataloging as custom tags and made searchable. You can search for data by name, size, and content.

You can use custom applications to extend the capabilities that are performed by IBM Data Cataloging.

- **Policy engine**
  Policies offer a method whereby you can schedule one-time or repetitive actions on a filtered set of records.
- **Applications**
  IBM Data Cataloging policies might contain applications in the actions parameters.

Policy engine

Policies offer a method whereby you can schedule one-time or repetitive actions on a filtered set of records.

The policy management API service is a RESTful web service that is designed to create, list, update, and delete policies. You can use a policy to initiate action on a select set of indexed documents or data. You can do a task immediately or on a set schedule.

Several types of policies that are supported by IBM Data Cataloging enrich the metadata records. You can create policies with information to determine which set of documents to run, the action to take, and when to run policies periodically.

A policy includes

Policy ID
: Name of the policy.

Filter
: Selects a set of documents to work.

Action
: ID, parameters, and schedule.

The following list is a description of the policies.

AUTOTAG
: A policy that tags a set of records based on filter criteria with a pre-defined set of tags.

CONTENT SEARCH
: A policy that uses the built-in content inspection capabilities of IBM Data Cataloging to extract content from source data and index it automatically into the IBM Data Cataloging catalog.

DEEP-INSPECT
: A policy that passes lists of files based on filter criteria to the analytics application that opens the source data file and extracts metadata information from it. The policy passes the data back to IBM Data Cataloging in the form of tags so you can do a search, and do the following activities:

  - Set up a filter to do a search query that finds the candidates to apply the policy.
    For example, you can set an action for filtered candidates AUTOTAG: `tag1: value`, `tag2: value`
  - Set a schedule to apply the policy by specifying the following methods:
    - Immediately
    - Periodically

Applications

IBM Data Cataloging policies might contain applications in the actions parameters.

Use an application when you want to do a specific action on data or metadata on IBM Data Cataloging.

You can define an application when you create a new DEEP-INSPECT policy. You can add parameters for an application during the process of creating a DEEP-INSPECT policy.

When you open the window for applications, you can see a view of a table with the following information:

Application
: The name of the application.

Parameters
: The parameters that were assigned to the application when the policy was created.

Action ID
: Actions that are supported by the application for enriching metadata. For example, CONTENTSEARCH, DEEP-INSPECT.

View or Delete
: Use the delete trashcan icon to remove the application from the database.

Graphical user interface

The IBM Data Cataloging graphical user interface is a portal that is used for running data searches, report generation, policy and tag management, and user Access Management. Based on a user's role, they might have access to one or more of these areas.

The IBM Data Cataloging environment provides access to users and groups. The role that is assigned to a user or group determines the functions that are available. Users and groups can also be associated with collections, which use policies that determine the metadata that is available to view.

User and group access can be authenticated by IBM Data Cataloging, an LDAP server, or the IBM Cloud® Object Storage System. The administrator can manage the user access functions.

Roles

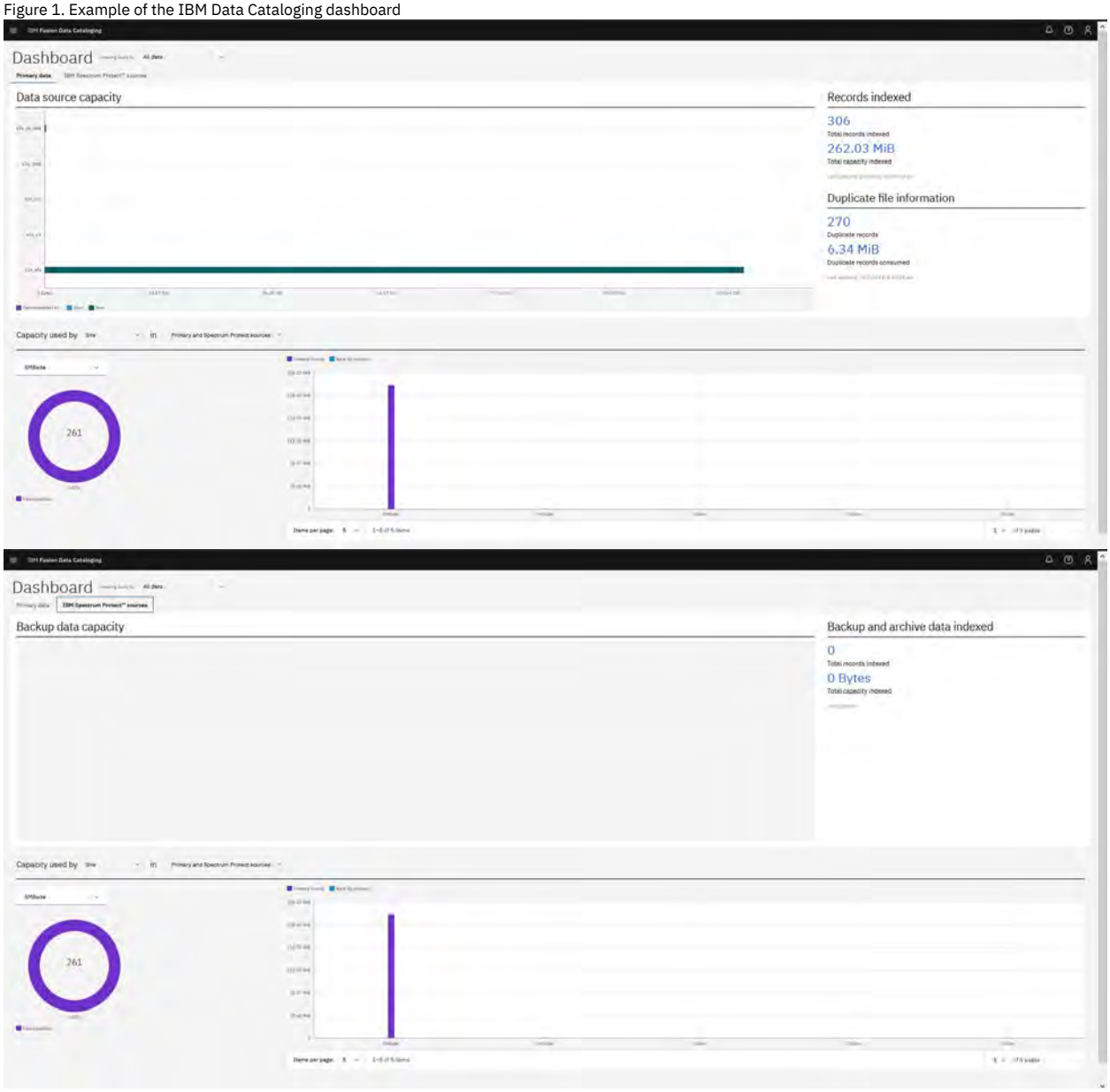
Roles determine how users and groups can access records on the IBM Data Cataloging environment.

If a user or group is assigned to multiple roles, the least restrictive role is used. For example, if a user is assigned a role of Data User, and is included in a data administrator role, the user has the privileges of a data administrator.

Dashboard

An example of the IBM Data Cataloging dashboard is shown.

Figure 1. Example of the IBM Data Cataloging dashboard



Data administrators and users can view the following:

- Metrics for the overall capacity used by every data source
- Total number of files
- Amount of capacity that is used by records with specific tags and facets, for example, owner, cluster, and size range
- Distribution of those records across data sources

Users can click any of the dashboard widgets to initiate a search and further explore and drill down into the data. Administrators and user can also perform the following:

- Monitor storage usage and data recommendations
- View total indexed data and capacity

- View duplicate file or object candidates. For example:
  - Number
  - Capacity used
- Preview capacity use by data facet. For example:
  - Classification
  - Owner
  - File type
- View data capacity by group or collection. For example:
  - Customer defined
  - Lab or project

Understanding size and capacity differences

IBM Data Cataloging collects size and capacity information:

- Size refers to the size of a file or object in bytes.
- Capacity refers to the amount of space the file or object consumes on the source storage in bytes.

For objects, size and capacity values always match. For files, size and capacity values can be different because of file system block overhead or sparsely populated files.

Note: Storage protection overhead (such as RAID values or erasure coding) and replication overhead are not captured in the capacity values.

Reports for IBM Data Cataloging

Reports for IBM Data Cataloging are grouped or non-grouped. Grouped reports have information for count and sum in columns and non-grouped reports have information in rows.

Data Curation Reports are a way for administrators, also known as data curators, to view the state of their storage environment in different ways. They can range from high-level grouped information to individual record level information.

For example, you can sort a report by owner, project, and department, or you can generate a list of records that meet a specific criterion. Additionally, you can create a report that lists the records in a project, not been reviewed for over a year. The owner of the data can evaluate whether to archive or delete the report.

For more information, see Reports*Reports* in *Data Cataloging: Administration Guide*.

For more information, see REST API for IBM Data Cataloging. For more information, see *REST API* in the *IBM Data Cataloging: REST API Guide*.

Installing IBM Data Cataloging

IBM Data Cataloging service is a modern metadata management software that provides data insight for exabyte-scale heterogeneous file, object, backup, and archive storage on premises and in the cloud. It can help you manage your unstructured data by reducing the data storage costs, uncovering hidden data value, and reducing the risk of massive data stores.

Before you begin

- Meet the system requirements to install the IBM Data Cataloging service.
- The following details are a base line for finding the resources that are needed for IBM Fusion IBM Data Cataloging service deployment. Based on the following tables, the resources can be estimated based on the number of approximate files that are required. The following are the resource values that are calculated per compute node: You must have at least two worker nodes, each with the same amount of resources available.
- IBM Fusion IBM Data Cataloging service must have dedicated compute resources. Ensure that you have enough to cover the resources limits to perform as expected:

Table 1. Profile requirements

CPU	RAM	Disk space	Network	Storage	Workload
77	162 GB	120 GB	10 GB	1 TB	1 B

- The standard deployment for IBM Data Cataloging service project requests and limits. Ensure you set the resource quota values for IBM Data Cataloging as follows:

```
kind: ResourceQuota
apiVersion: v1
metadata:
  name: dcs-resource-quota
  namespace: ibm-data-cataloging
spec:
  hard:
    limits.cpu: '72'
    limits.memory: 115Gi
    pods: '120'
    requests.cpu: '18'
    requests.memory: 26Gi
```

- Important: For the IBM Data Cataloging service to run successfully on all platforms, ensure that the storage classes have the following attributes:
  - ReadWriteMany (RWX) permissions
  - volumeBindingMode set to Immediate
  - AllowVolumeExpansion set to true
- If you have not configured the IBM operator catalog, then configure it. For the procedure to add IBM operator catalog, see Adding the IBM operator catalog.
- Go through troubleshooting information related to the installation of IBM Data Cataloging. See IBM Data Cataloging service installation and upgrade issues.
- IBM Data Cataloging can be deployed as containers on Red Hat OpenShift Virtualization Engine (OVE) but must run on normal worker nodes.

Procedure

1. Go to Services page in IBM Fusion user interface.
2. In the Available section, click Data cataloging tile.
3. In the Data cataloging window, go through the details of the service and click Install.
4. In the Install service message box, select a Storage class.
   Important:
   - If you want to use Global Data Platform as the storage provider, then it is recommended to select the default storage class **`ibm-spectrum-fusion`**. Otherwise, if you want to use Fusion Data Foundation, then select the **`ocs-storagecluster-cephfs`** storage class. You can also use a custom storage class that matches the requirements.
   - For deployments operating in FIPS mode, you must specify or select a **`storage class`** that uses **`encrypted storage`**.
     - If the default storage classes (**`ibm-spectrum-fusion`** or **`ocs-storagecluster-cephfs`**) are **`FIPS-compliant`** and use encrypted storage, you can continue to use them.
     - If not, you must configure and select a custom encrypted storage class that meets FIPS requirements.
     - For storage based on node SSDs, ensure the SSDs are encrypted at deployment time using encryption methods that meets FIPS requirements.
     - For storage based on Red Hat® OpenShift® Data Foundation, configure encrypted Red Hat OpenShift Data Foundation storage and use a storage class from that encrypted pool.
5. Click Install. In case of failures, go through the downloaded logs to understand the cause of the failure and fix the issue. For more information about service issues in IBM Fusion, see IBM Data Cataloging service installation and upgrade issues.
6. Validate the installation.
   - IBM Fusion user interface:
     After you enable the IBM Data Cataloging service, you can view the service version and health status. From the ellipsis menu, you can download logs and view documentation. After you successfully collect the logs, a success notification gets displayed. The notification disappears automatically after some time.

     Table 2. Health states IBM Data Cataloging service

State	Description
`Installing`	Service installation is in progress
`Upgrading`	Service upgrade is in progress
`Healthy`	Service is healthy
`Degraded`	Service is not healthy

   - Verify the IBM Data Cataloging installation from the OpenShift Container Platform web console:
     - Run the following command to verify the installed operators.

       ```
       oc -n ibm-data-cataloging get csv
       ```

       ```
       Expected output
       NAME                                          DISPLAY                           VERSION              REPLAC
       amqstreams.v2.3.0-1                           Red Hat Integration - AMQ Streams   2.3.0-1              amqstr
       db2u-operator.v3.1.0                          IBM Db2                           3.1.0
       ibm-spectrum-discover-operator.v210.0.0       IBM Storage Discover              210.0.0-1678324770
       ```

     - Run the following command to verify IBM Data Cataloging status.

       ```
       oc -n ibm-data-cataloging get isd
       ```

       ```
       Expected output
       NAME                              READY   SUMMARY                        ERROR   AGE   HEALTHSTATUS
       data-cataloging-service-instance   True    Awaiting next reconciliation           9d    Healthy
       ```

     - Run the following command to verify that all Persistent Volume Claims are bounded.

       ```
       oc -n ibm-data-cataloging get pvc
       ```

       ```
       Expected output
       NAME                        STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLA
       activelogs-c-isd-db2u-0     Bound    pvc-7e98eead-9a38-48b6-a880-16d730120796   150Gi      RWO            ocs-storag
       activelogs-c-isd-db2u-1     Bound    pvc-7af71786-1529-4b3c-bbd1-b5d1ee733d33   150Gi      RWO            ocs-storag
       c-isd-meta                  Bound    pvc-1ba20699-ddb8-4600-bf9f-4d637b14da49   50Gi       RWX            ocs-storag
       data-c-isd-db2u-0           Bound    pvc-488540db-c951-4f48-b65c-814bf634ca33   300Gi      RWO            ocs-storag
       data-c-isd-db2u-1           Bound    pvc-56a3105a-347e-4b55-b11a-d0a1d351d622   300Gi      RWO            ocs-storag
       data-isd-sasl-kafka-0       Bound    pvc-3924c9cc-173d-4244-90fa-a234325dc04b   100Gi      RWO            ocs-storag
       data-isd-sasl-zookeeper-0   Bound    pvc-3c66397b-ac03-4ed8-841a-e33d83355a23   64Gi       RWO            ocs-storag
       data-isd-ssl-kafka-0        Bound    pvc-ce0ebeaf-d2b6-4e40-8c31-e31d45a208e5   100Gi      RWO            ocs-storag
       data-isd-ssl-zookeeper-0    Bound    pvc-cd3fdd20-343b-49ca-ad1a-690f8eec3e60   64Gi       RWO            ocs-storag
       isd-backup                  Bound    pvc-2e077c8e-e992-4e1a-8b98-d24bfdbb5343   250Gi      RWX            ocs-storag
       isd-data                    Bound    pvc-b98e124f-1f98-43e5-9bf6-c8758c681192   250Gi      RWX            ocs-storag
       tempts-c-isd-db2u-0         Bound    pvc-7fdc9ac5-9158-48ac-bad3-3c24590f1f2b   50Gi       RWO            ocs-storag
       tempts-c-isd-db2u-1         Bound    pvc-a2adb668-a362-4f61-8ed3-ae103f9b2cd3   50Gi       RWO            ocs-storag
       ```

     - Run the following command to verify Db2u cluster detailed progress.

       ```
       oc -n ibm-data-cataloging get formations.db2u.databases.ibm.com isd -o go-template='{{range .status.components}}
       {{printf "%s,%s,%s\n" .kind .name .status.state}}{{end}}' | column -s, -t
       ```

       ```
       Expected output
       account                account-ibm-data-cataloging-isd  OK
       PersistentVolumeClaim  c-isd-meta                       OK
       secret                 c-isd-sshkeys-db2uhausr          OK
       secret                 c-isd-sshkeys-db2uadm            OK
       secret                 c-isd-sshkeys-db2instusr         OK
       secret                 c-isd-ldappassword               OK
       secret                 c-isd-ldapblueadminpassword      OK
       secret                 c-isd-instancepassword           OK
       secret                 c-isd-db2u-lic                   OK
       secret                 c-isd-certs-wv-rest              OK
       ```

```
secret                 c-isd-certs-db2u-api             OK
configmap              c-isd-db2uconfig                 OK
configmap              c-isd-db2regconfig               OK
configmap              c-isd-db2dbmconfig               OK
configmap              c-isd-db2dbconfig                OK
service                c-isd-ldap                       OK
Deployment             c-isd-ldap                       OK
service                c-isd-tools                      OK
Deployment             c-isd-tools                      OK
configmap              c-isd-db2u-api                   OK
service                c-isd-db2u-engn-svc              OK
service                c-isd-db2u-head-engn-svc         OK
Job                    c-isd-instdb                     OK
service                c-isd-etcd                       OK
StatefulSet            c-isd-etcd                       OK
service                c-isd-db2u-rest-svc              OK
networkpolicy          c-isd-rest-ext                   OK
Deployment             c-isd-rest                       OK
service                c-isd-db2u-graph-svc             OK
networkpolicy          c-isd-graph-ext                  OK
Deployment             c-isd-graph                      OK
service                c-isd-db2u-internal              OK
service                c-isd-db2u                       OK
StatefulSet            c-isd-db2u                       OK
networkpolicy          c-isd                            OK
networkpolicy          c-isd-ext                        OK
Job                    c-isd-restore-morph              OK
```

- Run the following command to verify the Db2u cluster CR status.

```
oc -n ibm-data-cataloging get db2ucluster
```

```
Expected output
NAME   STATE   MAINTENANCESTATE   AGE
isd    Ready   None               9d
```

- Run the following command to verify that the ports 50000 and 50001 are listing.

```
oc exec -n ibm-data-cataloging -it $(oc -n ibm-data-cataloging get po --no-headers --show-labels=true --selector
name=dashmpp-head-0 | awk '{print $1}') -- su - db2inst1 -c 'netstat -ntlp'
```

```
Expected output
Defaulted container "db2u" out of: db2u, init-labels (init), init-kernel (init)
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 10.254.4.11:60000       0.0.0.0:*               LISTEN      23707/db2sysc 0
tcp        0      0 10.254.4.11:60001       0.0.0.0:*               LISTEN      23712/db2sysc 1
tcp        0      0 0.0.0.0:9443            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:50022           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:50000           0.0.0.0:*               LISTEN      23707/db2sysc 0
tcp        0      0 0.0.0.0:50001           0.0.0.0:*               LISTEN      23707/db2sysc 0
tcp6       0      0 :::50052                :::*                    LISTEN      -
tcp6       0      0 :::50022                :::*                    LISTEN      -
```

- Run the following command to verify that workload is PUREDATA_OLAP.

```
oc exec -n ibm-data-cataloging -c db2u -it $(oc -n ibm-data-cataloging get po --no-headers --show-labels=true --
selector name=dashmpp-head-0 | awk '{print $1}') -- sudo su - db2inst1 -c "db2set -all | grep DB2_WORKLOAD= |
awk '{print \$2}'"
```

```
Expected output
DB2_WORKLOAD=PUREDATA_OLAP
```

- Run the following command to verify the database encryption configurations.

```
oc -n ibm-data-cataloging exec -c db2u $(oc -n ibm-data-cataloging get po --no-headers --show-labels=true --
selector name=dashmpp-head-0 | awk '{print $1}') -- su - db2inst1 -c "db2 get db cfg for bludb | grep 'Encrypted
database'"
```

```
 Encrypted database                                      = YES
```

- Run the following command to verify the Kafka CRs readiness.

```
oc -n ibm-data-cataloging get kafka
```

```
Expected output
NAME       DESIRED KAFKA REPLICAS   DESIRED ZK REPLICAS   READY   WARNINGS
 isd-sasl   1                        1                     True
 isd-ssl    1                        1                     True
```

- Run the following command to verify that the IBM® Spectrum Discover pods are running.

```
oc -n ibm-data-cataloging get pod -l component=discover
```

```
Expected output
 NAME                                      READY   STATUS      RESTARTS      AGE
isd-api-7564475d98-cs6nx                  1/1     Running     0             9d
isd-auth-5fcb8bd95f-jvzwp                 1/1     Running     0             9d
isd-backup-restore-689fc64f7-pgx22        1/1     Running     0             9d
isd-connmgr-7d4cdb4cc7-xnkzm              1/1     Running     1 (9d ago)    9d
isd-consumer-ceph-le-5f9b5d4676-4fsgg     1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-9k9cs     1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-bphqp     1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-cc2pz     1/1     Running     0             9d
```

```
isd-consumer-ceph-le-5f9b5d4676-chlvk        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-fchp5        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-jlbrw        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-l78cc        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-lpcsn        1/1     Running     0             9d
isd-consumer-ceph-le-5f9b5d4676-n2wqt        1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-54qm9         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-6hsq9         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-9p4p5         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-dhlgc         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-hr76h         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-mkvx8         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-tvscz         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-vrrd4         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-w59c2         1/1     Running     0             9d
isd-consumer-cos-le-846f5fd97f-xqncl         1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-4klcc       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-8r6w7       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-fqs5m       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-fwb8p       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-h78g5       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-lvwl5       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-nhznb       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-q218b       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-z6tfj       1/1     Running     0             9d
isd-consumer-cos-scan-79fb979585-zv2zw       1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-5mrrg      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-8rphj      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-b5dbv      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-b5lv2      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-d9gbm      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-f5rxf      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-m8nw6      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-ntfbb      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-ql55x      1/1     Running     0             9d
isd-consumer-file-scan-5c7565bf7b-qvggm      1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-6s75x    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-845bx    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-8kmnf    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-gntbt    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-mlnfb    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-p9lpt    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-rckm8    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-vkmpd    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-vmh8x    1/1     Running     0             9d
isd-consumer-protect-scan-67d5ffb65-zm4gf    1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-6dw9h       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-9nhsz       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-dnwpw       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-dz2mf       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-h2ffj       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-hdrxg       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-nf779       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-pf6n4       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-q8gqf       1/1     Running     0             9d
isd-consumer-scale-le-5d56449994-xzmvh       1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-49vkb     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-54pb8     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-5sr89     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-98lm9     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-d4zlj     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-fc82j     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-mnsn2     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-r2kc6     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-tglsz     1/1     Running     0             9d
isd-consumer-scale-scan-7bdc7957c6-zbhgv     1/1     Running     0             9d
isd-contentsearchagent-5d7bfbd887-r487x      1/1     Running     1 (9d ago)    9d
isd-db-schema-xm5cn                          0/1     Completed   0             9d
isd-db2whrest-6c6bbfcbc6-9v6v2               1/1     Running     1 (9d ago)    9d
isd-generate-license-6g94v                   0/1     Completed   0             9d
isd-importtags-64498dbdd8-5kqpb              1/1     Running     1 (9d ago)    9d
isd-keystone-7dc84b69f5-sw6nn                1/1     Running     0             9d
isd-license-check-27982200-c9gkm             0/1     Completed   0             2d21h
isd-license-check-27983640-ldxv2             0/1     Completed   0             45h
isd-license-check-27985080-rhzjl             0/1     Completed   0             21h
isd-policyengine-598f565867-zg5sn            1/1     Running     1 (9d ago)    9d
isd-producer-ceph-le-76b9885854-557b7        1/1     Running     0             9d
isd-producer-cos-le-85d446c9d8-zvzmg         1/1     Running     0             9d
isd-producer-cos-scan-d584f646d-2t669        1/1     Running     0             9d
isd-producer-file-scan-5586b48fd-zkt7q       1/1     Running     0             9d
isd-producer-protect-scan-7b66bb7c64-wz6qq   1/1     Running     0             9d
isd-producer-scale-le-8d7bc64b4-nvqbl        1/1     Running     0             9d
isd-producer-scale-scan-5df647956-qfl6p      1/1     Running     0             9d
isd-proxy-5f4d7dd8ff-44fwt                   1/1     Running     0             9d
isd-reports-c6kmq                            0/1     Completed   0             9d
isd-scaleafmdatamover-85cc44b566-644vx       1/1     Running     1 (9d ago)    9d
isd-scaleilmdatamover-6b86dd7b58-j28sn       1/1     Running     1 (9d ago)    9d
isd-sdmonitor-6fffc999b7-s2srd               1/1     Running     0             9d
isd-tikaserver-79548bdb4-ssft5               1/1     Running     0             9d
isd-ui-backend-f6c4b65d7-dwzrs               1/1     Running     0             9d
isd-ui-frontend-9659fc847-vcbmv              1/1     Running     0             9d
```

- Run the following command to verify the console route availability.

  ```
  oc -n ibm-data-cataloging get route console
  ```

```
Expected output
NAME      HOST/PORT                                          PATH   SERVICES   PORT    TERMINATION   WILDCARD
console   console-ibm-data-cataloging.apps.ocp2.vmlocal             isd-svc    <all>   reencrypt     None
```

Uninstalling IBM Data Cataloging

Steps to uninstall IBM Data Cataloging by using command line interface.

Procedure

1. Run the following command and export IBM Fusion namespace as an environmental variable.

   ```
   export FUSION_NS="<Fusion namespace>"
   ```

   Important: Replace `<Fusion namespace>` with your namespace.
2. Run the following command to prevent catalog source creation.

   ```
   oc -n ${FUSION_NS} patch fusionservicedefinition data-cataloging-service-definition --type='json' -p='[{"op": "replace",
   "path": "/spec/onboarding/serviceOperatorSubscription/triggerCatSrcCreate", "value":false}]'
   ```

3. Run the following command to scale down prerequisite operator.

   ```
   oc -n ${FUSION_NS} scale --replicas=0 deployment/isf-prereq-operator-controller-manager
   ```

4. Run the following commands to delete all the IBM Data Cataloging workload, networking and storage resources.

   ```
   oc project default
   export DCS_CR=$(oc get SpectrumDiscover -n ibm-data-cataloging -o jsonpath="{.items[*].metadata.name}")
   oc delete SpectrumDiscover ${DCS_CR} -n ibm-data-cataloging --timeout=2m --wait=true
   oc delete project ibm-data-cataloging --timeout=1m --wait=true
   ```

5. Run the following command to delete the IBM Fusion IBM Data Cataloging service instance object.

   ```
   oc -n ${FUSION_NS} delete fusionserviceinstance data-cataloging-service-instance
   ```

6. Run the following command to delete the `SecurityContextConstraints` objects.
   Important: The SCC objects are based on their namespace, and the namespace must be included in the name of the SCC object. The default namespace is `ibm-data-cataloging`, but you must change it if you use a different one.

   ```
   oc delete securitycontextconstraints/isd-scc-ibm-data-cataloging
   oc delete securitycontextconstraints/ibm-data-cataloging-c-isd-scc
   ```

7. Run the following command to delete the IBM Data Cataloging `ConsoleLink` object.

   ```
   oc delete consolelink/data-cataloging
   ```

8. Run the following command to delete IBM Data Cataloging `CustomResourceDefinitions` from the cluster.

   ```
   oc delete customresourcedefinition/spectrumdiscovers.spectrum-discover.ibm.com
   oc delete customresourcedefinition/spectrumdiscoverapplications.spectrum-discover.ibm.com
   ```

9. Run the following command to scale up prerequisite operator.

   ```
   oc -n ${FUSION_NS} scale --replicas=1 deployment/isf-prereq-operator-controller-manager
   ```

Configuring IBM Data Cataloging

This section provides information on how to deploy and configure IBM Data Cataloging capabilities.

- **Setting up IBM Data Cataloging custom user and password**
  Use this information to create a custom user and password for the IBM Data Cataloging service.
- **Configure data source connections**
  Data source connections describe the source data systems for which IBM Data Cataloging indexes metadata.
- **Editing and using the TimeSinceAccess and Size Range buckets**
  Users can group or aggregate data into three user-defined bucket ranges. The three user-defined bucket ranges are TimeSinceAccess, Size Range and FileGroup.
- **IBM Spectrum Storage software requirements**
  Use IBM Data Cataloging to index metadata from other applications and to orchestrate the data management.
- **IBM Data Cataloging installation with alternative VLAN**
  Install and configure IBM Data Cataloging that uses additional VLAN connected through the IBM Fusion upstream links.
- **Multiple connection managers**
  Multiple connection managers are a new capability that is designed to enhance scanning performance and enable parallel ingestion. It proves especially valuable in scenarios where data sources are geographically dispersed and need to be scanned as remote sources.
- **Enabling feature for skipping the snapshot directories**
  When the skipping feature is enabled, you can skip the metadata ingestion to the snapshot directories.
- **Managing IBM Data Cataloging rules when integrated with CAS**
  IBM Data Cataloging provides the ability to integrate with Content-Aware Storage to configure file ingestion filtering. This feature enables you to filter files from a data source based on specific criteria, such as file type or content, and ingest them into the corresponding domain.

Setting up IBM Data Cataloging custom user and password

Use this information to create a custom user and password for the IBM Data Cataloging service.

About this task

It is required to perform this procedure before installation to prevent errors caused by the user being already initialized with the default values.

Procedure

1. Run the following command to create the **ibm-data-cataloging** namespace.

   ```
   oc create namespace ibm-data-cataloging
   ```

2. Run the following command to create the keystone secret to hold the custom user and password.

   ```
   oc -n ibm-data-cataloging create secret generic keystone --from-literal=user=<CUSTOM_USER> --from-literal=password=
   <CUSTOM_PASSWORD>
   ```

3. Deploy the IBM Data Cataloging service from theIBM Fusion user interface.

Configure data source connections

Data source connections describe the source data systems for which IBM Data Cataloging indexes metadata.
Creating data source connections in IBM Data Cataloging identifies source storage systems that are to be indexed by IBM Data Cataloging.

For some data source types, a network connection is (optionally) created to allow for automated scanning and indexing of the source system metadata. IBM Data Cataloging will not index data from unknown sources, so creating a data source connection is the first step towards cataloging any source storage system.

Remember: Depending on when the scan is stopped, stopping a running scan might result in an inconsistent database state for the connection.
You can add data connections to the source storage systems from the IBM Data Cataloging graphical user interface and REST API. For more information on configuring data source connections offline, see Configuring data source connections offline. *Configuring data source connections offline* in the *IBM Data Cataloging: Concepts, Planning, and Deployment Guide*.

IBM Data Cataloging discards any data that comes in from an unknown connection. Therefore, connections must be established before data ingestion. To see the list of defined connections, use the Connections tab under the Data source management window of the GUI.
Remember: If you use a MAC, you might have to adjust the scroll bar settings in System Settings to see all available connection types. For example, activate the Show scroll bars: Always option.
Typically, a data source is equivalent to a single file system or object vault or bucket. A data source connection is an alias for the combination of a cluster name and a data source within the cluster. This allows multiple file systems or buckets or vaults with the same name to be indexed by IBM Data Cataloging when they are in separate clusters.

Remember:

- IBM Data Cataloging does not support file or file path names that use characters that are not part of the UTF-8 character set.
- After you do a manual scan, run a Metadata summarization database refresh:
  1. In Data source management > Metadata summarization database, go to Data connections > Discover information.
  2. Click refresh and wait for the process to complete successfully. The expected time of completion is based on the amount of records to process. For example, 30 minutes.

- **IBM Storage Scale data source connections**
  You can create an IBM Storage Scale data source connection, scan a data source, and manually initiate a scan.
- **IBM Storage Archive data source connections**
  You can define tags and policies in IBM Data Cataloging based on values that are derived from IBM Spectrum® Archive metadata to help in searching and categorizing files.
- **IBM Cloud Object Storage data source connection**
  You can create the IBM Cloud® Object Storage (COS) connection and initiate a scan.
- **IBM Data Cataloging and S3 object storage data source connections**
  Use this information to understand how IBM Data Cataloging works with S3-compliant object store.
- **Scanning an Elastic Storage Server data source connection**
  Use the IBM Data Cataloging GUI to scan an Elastic Storage Server (ESS) data connection.
- **Creating a Network File System data source connection**
  You can use the IBM Data Cataloging graphical user interface to create a Network File System (NFS) data connection.
- **Creating an SMB data source connection**
  Creating an SMB data connection by using the IBM Data Cataloging graphical user interface.
- **Creating an IBM Storage Protect data source connection**
  Creating the IBM Storage Protect data connection by using the IBM Data Cataloging graphical user interface.
- **Configuring data source connections offline**
  Multiple source data connections can be added to the IBM Data Cataloging system through offline mode.

IBM Storage Scale data source connections

You can create an IBM Storage Scale data source connection, scan a data source, and manually initiate a scan.

Tip: If the data source connection is used by the ScaleILM application to tier data by using IBM Spectrum® Archive, the `host` setting of the IBM Storage Scale connection must specify one of the IBM Storage Archive nodes. For more information, see Tiering data by using ScaleILM application.For more information, see the topic *Tiering data by using ScaleILM application* in the *Data Cataloging: Administration Guide*.
IBM Data Cataloging supports the following method of scanning IBM Storage Scale data sources:

Automated scanning
: A data source connection is defined on IBM Data Cataloging, including details of how to connect to the IBM Storage Scale system. IBM Data Cataloging connects to the IBM Storage Scale system by using these details, scans the file system and sends the details back to IBM Data Cataloging. For more information, see Prerequisites for automated scanning.For more information, see the topic *Prerequisites for automated scanning* in the *Data Cataloging: Administration Guide*.

- **IBM Storage Scale scanning considerations**
  The following sections comprise considerations that you need to understand to scan IBM Storage Scale effectively.
- **Prerequisites for automated scanning**
  You can use IBM Storage Scale automated scanning features.
- **Creating an IBM Storage Scale data source connection**
  You can use the IBM Data Cataloging graphical user interface to create data connections from the source storage systems.
- **Automated scanning of an IBM Storage Scale data source**
  As an administrator, you can initiate an IBM Storage Scale scan from IBM Data Cataloging to collect system metadata from IBM Storage Scale file system.
- **Manual scanning of an IBM Storage Scale data source**
  How to configure IBM Data Cataloging to connect to IBM Spectrum Scale.

IBM Storage Scale scanning considerations

The following sections comprise considerations that you need to understand to scan IBM Storage Scale effectively.

- **Security considerations**
  Use this information to securely scan a system connection.
- **Performance considerations**
  Use this information to scan a system connection without degrading performance.

Security considerations

Use this information to securely scan a system connection.

Scanning an IBM Storage Scale instance involves the use of the **mmapplypolicy** command on the IBM Storage Scale system, which requires superuser permissions. When you are creating the data source connection for the target IBM Storage Scale system in the IBM Data Cataloging interface, you are prompted for a *userid* and *password* to enable automated scans. You are not required to provide these credentials if scans are run only manually on the target IBM Storage Scale system by an administrator. However, if you want to run automation and/or schedule scans, then the authentication credentials are required. By default, IBM Data Cataloging uses password authentication to the Scale cluster to run commands remotely. However, you can supply your own RSA private key by selecting the shared key authentication option when you are configuring the connection if you want to avail passwordless authentication.

Rather than providing root login credentials, an administrator must create a special user ID with limited permissions on the IBM Storage Scale system. The administrator must also enable a password-less **sudo** for the user ID, to the binaries needed for scanning. This prevents someone from gaining root access to the target IBM Storage Scale system if the IBM Data Cataloging system is somehow compromised.

Changing passwordless SSH keys

You can rotate RSA authentication key pairs for passwordless SSH on a frequency and remove old security keys from the authorized_hosts file on the IBM Storage Scale node that IBM Data Cataloging connects to. To update the authentication keys, follow these steps:

1. Ensure that the id_rsa.pub contents for the new authentication key pair are in the ~/.ssh/authorized_hosts file for the user that is specified in the IBM Data Cataloging connection document for the IBM Storage Scale target file system.
2. Edit the connection and paste the contents of the new private key file (id_rsa) in the input form.

After you edit the connection with the new private key file, IBM Data Cataloging uses it to connect to the IBM Storage Scale target system.

Performance considerations

Use this information to scan a system connection without degrading performance.

Running a scan policy on an IBM Storage Scale system can be resource intensive and cause noticeable performance degradation on the IBM Storage Scale system. Often, system administrators choose to designate certain nodes or node classes for running the scans. The IBM Data Cataloging interface has an input field when creating IBM Storage Scale connections for the administrator to specify which nodes or node class(es) they would like to run the scan on. The value `all` will run the scan across all nodes in the cluster. Any other list (comma separated) will be treated as a list of nodes or node classes on which to run the scan. Scan times vary by the size of the filesystem, how many nodes are used in the scan, how many CPUs are used per node, and whether or not the IBM Storage Scale cluster metadata is in flash memory.

Prerequisites for automated scanning

You can use IBM Storage Scale automated scanning features.

IBM Data Cataloging supports two levels of automated scanning of IBM Spectrum Scale systems. Both levels require that IBM Data Cataloging must establish a password-less Secure Shell (SSH) connection to the IBM Storage Scale clustered system that is being scanned.

The difference between the two levels lies in the manner in which the output is handled. The factors that are considered inspects whether:

- The output of the IBM Storage Scale policy that is run to do the scan is stored in a file (which then must be automatically copied back to IBM Data Cataloging and ingested locally); Or
- If the output is, instead, pushed to the ingest Kafka queue of IBM Data Cataloging system directly from the IBM Storage Scale policy output.

The Kafka queue of IBM Data Cataloging system is more space-efficient and time-efficient but has certain dependencies on the IBM Storage Scale clustered system that must be met so that it can function. The IBM Data Cataloging automated scan code determines whether the dependencies are met on the IBM Storage Scale clustered system.

If the dependencies are met on the IBM Storage Scale clustered system, it attempts to scan the system by using the optimized path. If the dependencies are not met on the IBM Storage Scale clustered system, it defaults to the file copy path.

The following sections list the prerequisites for creating a connection and performing automated scanning. These prerequisites consider security and performance factors. For more information, see IBM Storage Scale scanning considerations .

- **Identify the IBM Storage Scale cluster and node list**
  Identify a node to connect with the GPFS cluster.
- **Creating or identifying a user ID and password for scanning**
  Identify a user to perform scanning or create a new user ID.
- **Validate scan user permissions and configuration**
  Ensure that the user has the required permissions and configurations for scanning.
- **Check the dependencies for optimized scanning**
  Identify the dependencies that must be met on the IBM Storage Scale system to optimize automated scan ingest.

Identify the IBM Storage Scale cluster and node list

Identify a node to connect with the GPFS cluster.

Identify a node in the target IBM Storage Scale cluster to use for the IBM Data Cataloging connection to the GPFS cluster.

You must identify the node list or node class that participates in the scanning activity.

Creating or identifying a user ID and password for scanning

Identify a user to perform scanning or create a new user ID.

About this task

You can identify an existing user to perform scanning or follow these steps on the IBM Storage Scale system to create a special user ID for scanning.

Procedure

1. Log in to the IBM Storage Scale management node as **root**.
   Alternatively, you can **sudo** to root from another user ID.
2. Use the following `adduser` steps to ensure that you are able to ssh into the cluster:
   a. `adduser` <user> `-m`
   b. `passwd` <user>
3. Run: **visudo**.
   a. Add the following line in the users section:
   <user> `ALL=NOPASSWD: /usr/lpp/mmfs/bin/mmapplypolicy, /usr/lpp/mmfs/bin/mmrepquota`
   b. Write and quit: `:wq`
4. Create an IBM Data Cataloging working directory and ensure that <user> has write permissions.
   For example: `mkdir -p /gpfs/fs1/sd_scan -m 770;`
   `chown <user> /gpfs/fs1/sd_scan`
5. To execute a Data Mover policy for tiering data on the IBM Storage Scale cluster by using the ScaleILM application on the IBM Storage Scale connection , do the following:
   a. Run: **visudo**
   b. Add or update the following line in the users section:
   <user> `ALL=NOPASSWD: /usr/lpp/mmfs/bin/mmapplypolicy, /usr/lpp/mmfs/bin/mmrepquota, /usr/lpp/mmfs/bin/mmlspool, /opt/ibm/ltfsee/bin/eeadm,/usr/bin/ls`
   c. Write and quit: `:wq`
   For more information, see Tiering data by using ScaleILM application. For more information, see the topic *Tiering data by using ScaleILM application* in the *Data Cataloging: Administration Guide*.
6. To execute a Data Mover policy for copying data to the IBM Storage Scale cluster by using the ScaleAFM application on the IBM Storage Scale connection, do the following:
   a. Run: **visudo**.
   b. Add or update this line in the users section:
   `<user>ALL=NOPASSWD: /usr/lpp/mmfs/bin/mmapplypolicy, /usr/lpp/mmfs/bin/mmrepquota, /usr/lpp/mmfs/bin/mmlspool, /usr/lpp/mmfs/bin/mmafmctl, /usr/lpp/mmfs/bin/mmafmcosctl, /usr/lpp/mmfs/bin/mmaddcallback, /usr/lpp/mmfs/bin/mmdelcallback`
   c. Write and quit: `:wq`

For more information, see Copying data using ScaleAFM application. For more information, see the topic *Copying data using ScaleAFM application* in the *Data Cataloging: Administration Guide*.

Validate scan user permissions and configuration

Ensure that the user has the required permissions and configurations for scanning.

You need to ensure that it is possible to use Secure Shell (SSH) to log in to the IBM Storage Scale system with the scanning user ID and password.

Run the **sudo /usr/lpp/mmfs/bin/mmapplypolicy**.

You also need to validate the listed factors for the scanning-related working directory:

- It exists.
- It is globally accessible by the scan worker nodes.
- The scan user has write permissions to the directory (ownership of the directory is preferred but not mandatory).

Place the **id_rsa** and **id_rsa.pub** files in **/opt/ibm/metaocean/data/connections/scale/** directory on the IBM Data Cataloging instance if a specific RSA key pair for password-less SSH is wanted.

You need to validate if the listed things are installed on the IBM Storage Scale node. This node is identified while you are identifying a node in the target IBM Storage Scale cluster. The node in the target is identified for use in the IBM Data Cataloging connection to the GPFS cluster:

- A python2 level of at least Python 2.7.5
- A sufficient level of librarian
- An appropriate level of confluent-Kafka

Note: This validation is recommended for optimized scan ingestion. However, it is optional and can be skipped.

Check the dependencies for optimized scanning

Identify the dependencies that must be met on the IBM Storage Scale system to optimize automated scan ingest.

The dependencies that must be satisfied on the IBM Storage Scale system to optimize automated scan ingest from IBM Data Cataloging are:

- A `librdkafka` library 0.11.4 or later
- A Python 3.0 or later with accompanying Python package installer (pip3)
- A `confluent-kafka` version that is greater than or equal to the installed `librdkafka` version.

If these dependencies are met, the scan output is pushed to the ingest Kafka queue of IBM Data Cataloging system directly from the IBM Storage Scale policy output.

An administrator can determine whether `librdkafka` is installed on the IBM Storage Scale node by running the **find /usr -name "*librdkafka*"** or **ls /lib64/librdkafka*** commands. The `librdkafka` package is included with newer levels of IBM Storage Scale on x86 and ppc64le platforms. However, it can also be built from the source code on older levels of IBM Storage Scale or ppc64 platforms. If the IBM Storage Scale system runs on Red Hat® Enterprise Linux® (RHEL) and is connected to a Red Hat Satellite, you can install it by running the Yellowdog Updater Modified (YUM) command **yum install librdkafka** as **root**. You can find source packages of `librdkafka` here: https://github.com/edenhill/librdkafka

The user ID specified in the data source connection must be able to locate the following two binaries by using the OS shell path:

1. A Python 3 binary as either **python** or **python3**
2. A Python package installer as **pip3**
   Note: Symbolic links or aliases may be used to locate the Python executables.

After you install a sufficient version of Python, you can install `confluent-kafka` by using pip. To get pip, you must install the `python-setuptools` package, which provides a binary called `easy_install`. For more information, see https://pypi.org/project/setuptools/#files

After `easy_install` is available, you can install pip by running **easy_install-2.7** pip as **root**. After you install pip, you can install `confluent-kafka` by running **pip install confluent-kafka** as **root**.

Creating an IBM Storage Scale data source connection

You can use the IBM Data Cataloging graphical user interface to create data connections from the source storage systems.

Procedure

1. Log in to the IBM Data Cataloging web interface with a user ID that has the Data Admin role that is associated with it.
   The data admin access role is required for creating connections. For more information, see Managing user access. For more information, see *Managing User Access* in the *Data Cataloging: Administration Guide*.
2. Click  menu and go to Data connections > Connections to display the different types of data source connection names, connection type, clusters, data source, site, state, scan status, next scan, and Add Connection button.
3. Click Add connection to display a new window that shows Add data source connection.
   You can enter in the connection name and connection type. The connection types are:
   - IBM Storage Scale

- IBM Cloud® Object Storage
- Network File System (NFS)

4. Complete the following steps:
   a. In the field for Connection name, define a Connection name.
   b. Choose type of connection from Connection type drop-down list.
5. Set the connection type to IBM Storage Scale. The page displays the connection name, user, password, working directory, and scan directory information that you can enter. You can also schedule a data scan, select a collection, or enable live events.
   If you click Enable live events, you can enable the IBM Storage Scale watch folder on the specified file system.
6. Complete updating values for all the fields to add the IBM Storage Scale connection type, and click Submit Connection.

   For IBM Storage Scale connections, you can enter the following information:

   Connection name
   : The name of the connection, an identifier for the user. For example, filesystem1.
   Note: It must be a unique name within IBM Data Cataloging.

   User
   : A user ID that has permissions to connect to the data source system and initiate a scan.

   Password
   : The password for the user ID specified in user.

   Authentication Type
   : The password authentication can be done by using the password that is provided to authenticate with the IBM Storage Scale cluster. The shared RSA key authentication performs a password less authentication by using a private key that is provided by the system administrator and whose public key exists in the authorized keys for the specified user on the Scale host.

   Note: IBM Data Cataloging 2.0.3.1 removes the support for self-generated RSA key pair for IBM Data Cataloging. Any existing connections that use that method is updated to password based authentication and the self-generated key pair is removed during the upgrade to 2.0.3.1 or later. If the password for the scan user that is stored in IBM Data Cataloging is no longer valid, that can result in scan failures after the update. To rectify this, you must edit the connection and provide a valid password for the scan user or a valid RSA private key for authentication.

   Working Directory
   : A scratch directory on the source data system where IBM Data Cataloging can put its temporary files.
   Note: When you edit an existing connection and change the User from a root user to a non-root user, you must also change the Working Directory. This change is necessary because the non-root User cannot access the files that are previously created by the root user in the existing Working Directory.

   Scan Directory
   : The root directory of the scan. All files and directories under this directory are scanned. Typically, this directory is the base directory of the file system. For example, /gpfs/fs1.

   Connection Type
   : The type of source storage system this connection represents.

   Site
   : An optional physical location tag that an administrator can provide to see the physical distribution of their data.

   Cluster
   : The IBM Storage Scale or GPFS cluster name. To obtain, run the following command from the IBM Storage Scale file system: /usr/lpp/mmfs/bin/mmlscluster.

   Host
   : The hostname or IP address of an IBM Storage Scale node from which a scan can be initiated, for example a quorum-manager node.

   File system
   : The short name (omit /dev/) of the file system to be scanned. For example, fs1.
   Note: It is important to exactly match the file system name (data source) that IBM Storage Scale populates in the scan file. Run the following command on the IBM Storage Scale system: `/usr/lpp/mmfs/bin/mmlsmount all`

   Node list
   : The comma-delimited list of nodes or node classes that participates in the scan of an IBM Storage Scale file system. For example, `scale01,scale02`.

   ```
   Node  Daemon node name  IP address    Admin node name  Designation
   -------------------------------------------------------------------
      1   msys111-10g       172.16.8.111  msys111-dmz      quorum-manager-perfmon
      2   msys112-10g       172.16.8.112  msys112-dmz      quorum-manager-perfmon
      3   msys113-10g       172.16.8.113  msys113-dmz      quorum-manager-perfmon
   ```

   Note: When you create data source connections for IBM Storage Scale file systems, it is important to exactly match the cluster name and the file system name (data source) that IBM Storage Scale populates in the scan file.
   Run the following commands on the IBM Storage Scale system.
   a. Run the following command to display information about the GPFS cluster:

   ```
   $ /usr/lpp/mmfs/bin/mmlscluster

   GPFS cluster information
   ========================
    GPFS cluster name:         modevvm19.tuc.example.com,
    GPFS cluster id:           7146749509622277333
    GPFS UID domain:           modevvm19.tuc.example.com
    Remote shell command:      /usr/bin/ssh
    Remote file copy command:  /usr/bin/scp
    Repository type:           CCR
   Node  Daemon node name           IP address    Admin node name            Designation
   ------------------------------------------------------------------------------------------
   1     modevvm19.tuc.example.com  203.0.113.24  modevvm19.tuc.example.com  quorum-manager
   ```

   b. Run the following command to display information about file systems that are mounted: on the nodes:

   ```
   $ /usr/lpp/mmfs/bin/mmlsmount all
   File system gpfs0 is mounted on 1 nodes
   File system Data_Science_8M is mounted on 7 nodes.
   File system icp4D_data_fs_master1 is mounted on 8 nodes.
   File system icp4D_data_fs_master2 is mounted on 8 nodes.
   File system icp4D_data_fs_master3 is mounted on 8 nodes.
   ```

Automated scanning of an IBM Storage Scale data source

As an administrator, you can initiate an IBM Storage Scale scan from IBM Data Cataloging to collect system metadata from IBM Storage Scale file system.

Before you begin

You can include or exclude the files during initial IBM Storage Scale scan process by configuring the following environment variable:

INCLUDE_SCALE_SNAPSHOTS
: When the *INCLUDE_SCALE_SNAPSHOTS* variable value is set to 'false' (default value), the IBM Storage Scale scan excludes all the files that are inside the .snapshots directories, otherwise, if the variable value is set to 'true', the scan includes all the files, including the .snapshots directories.

About this task

To set the *INCLUDE_SCALE_SNAPSHOTS* variable by using configmap, see Enabling skip snapshot directories feature on Red Hat® OpenShift®*Enabling skip snapshot directories feature on Red Hat® OpenShift* in the *IBM Storage Scale: Administration Guide*.

When a scan is initiated from the IBM Data Cataloging graphical user interface, the data moves asynchronously back to the IBM Data Cataloging.
Remember: Before you initiate a scan, see IBM Storage Scale scanning considerations.
Automated scanning and data ingestion relies on an established and active network connection between the IBM Data Cataloging instance and the source IBM Storage Scale management node. If the connection cannot be established, the state of the data source connection displays `'unavailable'` and the option for automated scanning does not appear in the IBM Data Cataloging GUI for that connection.

Note: You cannot run scans unless you add override warnings in the configuration file.

Procedure

1. Log in to IBM Data Cataloging web interface.
2. Click  menu and go to Data connections > Connections.
3. Select the data source connection name that you want to scan. Ensure that the connection is online for your system ready to scan. (There is an indicator in the State column.)
4. Select Scan now to start the scan, and a small message appears to confirm that the connection name you specify is being scanned.
   You can view the status of the scan on the table in the Scan Status column for the target connection. After the Scan Status has a check mark next to it, the scan is complete.
   Remember: You can also specify a time to begin the scan. Any time zones specified default to Coordinated Universal Time (UTC) time. So, if you specify your scan for 12 noon, it is 12 noon in UTC.

- **Automated scanning of an IBM Storage Scale fileset**
  As an administrator, you can initiate the IBM Storage Scale scan from IBM Data Cataloging to collect system metadata from the IBM Storage Scale file set or file sets.

Automated scanning of an IBM Storage Scale fileset

As an administrator, you can initiate the IBM Storage Scale scan from IBM Data Cataloging to collect system metadata from the IBM Storage Scale file set or file sets.

Before you begin

This feature adds a requirement for non-root user IDs that are used for scanning IBM Storage Scale data source systems. This feature uses the **mmlsfileset** command to retrieve the list of available file sets from the target system when you have root-level permissions. So, if you use a non-root user ID it must have sudo access to **mmlsfileset** for this function to work.

There is already a requirement for a non-root scan user to have sudo access to **mmapplypolicy**, so this requirement adds **mmlsfileset** as an extra required command.
Note: You cannot query the available file sets on a target IBM Storage Scale connection or initiate a file set level scan unless you fulfill this requirement.

About this task

Scan the IBM Storage Scale file set or file sets to insert or update the records for the files that are found by IBM Data Cataloging in that file set or file sets. The scan is scoped to the specified file set, which ensures a faster total scan than scanning the entire file system. Multiple file sets can be specified in a single scan operation, but the scanning of each file set is done successively.

As the scan progresses, the status message is updated to indicate the following information:

- The status message indicates which file set is being scanned.
- The status message indicates when data operations (such as transferring files or indexing data) occur.

This status message can be seen in the GUI on the data source connections table or it can be queried by using the REST API.
This feature works irrespective of whether the data is returned to IBM Data Cataloging by using a direct Kafka connection or by using the file copy method. After a file set level scan completes, a scan generation is recorded or committed.

Additionally, an internal reclamation policy is generated to remove any deleted files that did not appear in the updated scan. The scope of this reclamation policy is limited to the file set that is scanned and does not affect other file sets or the actual file system. This limitation helps you achieve consistency with the source IBM Storage Scale system at file set level granularity.

Procedure

1. Go to the IBM Data Cataloging GUI.
2. Click  menu and go to Data connections > Connections table.
   Select the data source connection name and click Scan now, which opens the Select scan type dialog box.
   You can select whether to scan the entire file system or to scan a list of file sets.
   Important: Connection types other than IBM Storage Scale and SMB/CIFS do not open this dialog box. Additionally, Scan now continues to function as it has, which means that there is an immediate initiation of a full connection scan.
3. Select either Scan all to scan all file sets or Select Filesets to scan a specific file set.
   Selecting Scan all initiates a full scan of the file system. If you choose to scan all file sets, click Scan to run the scan.

   Selecting Select Filesets initiates a specific file scan. Click Next to open the Select Individual Filesets dialog box. Use this dialog box to select the specific file sets that you want to scan. Search the table by using the table search header:
   a. You can select file sets by clicking the row of the table that represents that file set. Clicking the row highlights that row and the count under View X selected filesets increases by 1.
   b. You can also select file set by filtering the search criteria. The table can be filtered to show only the selected file sets by clicking View X selected filesets, for ease of review. For example, you can enter `fs` to display all file set with those characters in that order. Click the file set in the table row that you want to select to run the scan on that file set.

   To go back to viewing all available file sets, click View X selected filesets again. The button changes to View all filesets when you view only the selected file sets.
4. After you select all wanted file sets, you can initiate the scan by clicking Scan. Clicking Scan takes you to the Connections table.
   A notification indicates when the scan starts (or that the scan fails if there is a problem). You can view the status of the scan on the table in the Scan Status column for the target connection.
   Remember: After the Scan Status has a check mark next to it, the scan is complete for all selected file sets.

Manual scanning of an IBM Storage Scale data source

How to configure IBM Data Cataloging to connect to IBM Spectrum Scale. After completing these steps, data can be ingested from an IBM Spectrum Scale data source to IBM Data Cataloging for metadata indexing.

Before you begin

Create the data source connection to IBM Storage Scale. For more information, see Configure data source connections.
You can include or exclude the files during initial IBM Storage Scale scan process by configuring the following environment variable:

INCLUDE_SCALE_SNAPSHOTS
: When the *INCLUDE_SCALE_SNAPSHOTS* variable value is set to 'false' (default value), the IBM Storage Scale scan excludes all the files that are inside the .snapshots directories, otherwise, if the variable value is set to 'true', the scan includes all the files, including the .snapshots directories.

To set the *INCLUDE_SCALE_SNAPSHOTS* variable by using configmap, see Enabling skip snapshot directories feature on Red Hat® OpenShift®*Enabling skip snapshot directories feature on Red Hat® OpenShift* in the *IBM Storage Scale: Administration Guide*.

The minimum connection parameters required for manual scanning are:

- Connection Name
- Connection Type
- Cluster
- Filesystem

Restriction: IBM Data Cataloging uses a unit separator (ASCII code 0x1F) as the field delimiter for ingestion into the database. This means that data which contains this character in path/file/object names results in improper parsing of the input data and the records are rejected by IBM Data Cataloging.

Procedure

1. Perform a file system scan to collect system metadata from IBM Spectrum Scale to be ingested into IBM Data Cataloging. For more information, see Performing file system scan to collect metadata from IBM Storage Scale.
2. Copy the output of the file system scan to the IBM Data Cataloging master node. For more information, see Copying the output of the IBM Storage Scale file system scan to the IBM Data Cataloging master node.
3. Ingest data from the file system scan in IBM Data Cataloging. For more information, see Ingesting metadata from IBM Storage Scale file system scan in IBM Data Cataloging.
4. Ingest quota information from the file system. For more information, see Ingesting quota information from the file system.

Performing file system scan to collect metadata from IBM Storage Scale

You can use the file system scanning tool, IBM Storage Scale Scanner, to collect system metadata from IBM Storage Scale to be ingested into IBM Data Cataloging.

About this task

The IBM Storage Scale Scanner tool uses the IBM Storage Scale information lifecycle management (ILM) policy engine to obtain the system metadata about the files stored on the file system. The system metadata is written to a file, which is then transferred to the IBM Data Cataloging master node. The file is then ingested within the node and analytics is carried out to provide search, duplicate file detection, archive data detection, and capacity show-back report generation. The following system metadata is collected from the file system scan:

Key name	Description

Key name	Description
Site	The site where the file or object resides.
Platform	The source storage platform that contains the file or object.
Size	The size of the file.
Owner	The owner of the file.
Path	The subdirectory where the data resides.
Name	The name of the data.
Permissions	The permissions for the file (mode).
ctime	The change time of the file metadata (inode).
mtime	The time when the data was last modified.
atime	The time when the data was last accessed.
Filesystem	The name of the IBM Storage Scale file system that is storing the data.
Cluster	The name of the IBM Storage Scale cluster.
inode	The IBM Storage Scale inode that is storing the data.
Group	The Linux® group associated with the file.
Fileset	The file set that stores the file.
Pool	The storage pool where the file resides.
Migstatus	If applicable, indicates whether the data is migrated to tape or object.
migloc	If applicable, indicates the location of the data if migrated to tape or object.
ScanGen	Scan generation - useful to track rescans.

The IBM Storage Scale Scanner tool also collects quota information by calling **mmrepquota**.

The tool comprises the following files:

- scale_scanner.py: The tool that starts the IBM Storage Scale ILM policy.
- scale_scanner.conf: The configuration file used to customize the behavior of the scale_scanner.py tool.
- createScanPolicy: The script that is called internally by the tool.

Procedure

Install the IBM Storage Scale Scanner tool by unpacking the utility from the IBM Data Cataloging node to the required location on the IBM Storage Scale cluster node.

1. Log in to the IBM Data Cataloging node through Secure Shell (SSH) with the `moadmin` username and password:

   ```
   ssh modadmin@data.cataloging.ibm.com
   ```

2. Change to the directory that contains the IBM Storage Scale scanning utility

   ```
   /opt/ibm/metaocean/spectrum-scale
   ```

3. Copy the createScanPolicy, _init_.py, scale_scanner.conf, and scale_scanner.py files to a node in the IBM Storage Scale cluster:

   ```
   scp * root@spectrumscale.ibm.com:/my_scanner_directory

   createScanPolicy 100% 3320 3.2KB/s 00:00
   init.py 100% 427 0.4KB/s 00:00
   scale_scanner.conf 100% 1595 1.6KB/s 00:00
   scale_scanner.py 100% 13KB 13.2KB/s 00:00
   ```

4. On the IBM Storage Scale node where you install the scanning utility, edit the configuration file (scale_scanner.conf) as follows:
   a. Use the IBM Data Cataloging UI to create a connection to the SS system on which you start a manual scan for. Set the **filesystem** and **scandir** fields, and optionally set the **outputdir** and **site** fields in the [spectrumscale] stanza of the file.

      ```
      [spectrumscale]
      # Spectrum Scale Filesystem which hosts the scan directory
      # example:  /dev/gpfs0
      filesystem=/dev/gpfs0
      # The directory path on Spectrum Scale Filesystem to perform scan on
      # example: /gpfs0
      # specifies a global directory to be used for temporary storage during
      # mmapplypolicy command processing. The specified directory must be
      #mounted with read/write access within a shared file system
      mountpoint=mount point of the gpfs filesystem
      # It is unclear what the mount_point should be, but setting the mount point
      # to the mount point of the scale file system on the IBM Spectrum Scale node works.
      scandir=/gpfs0
      # The directory to store output data from the scan in (default is
      # scandir)
      outputdir=
      # The site tag to specify a physical location or organization identifier.
      # If you use this field, remove the comment (#)
      #site=
      ```

   b. Set the **scale_connection**, **master_node_ip**, and **username** fields in the **[datacataloging]** stanza of the file.
      Note: **scale_connection** refers to the name of the IBM Storage Scale file system that is scanned and ingested into IBM Data Cataloging. The **scale_connection** value must match the value that is defined in the **Data Source** column of the Data Connections page in the IBM Data Cataloging GUI. The username must be a valid name of the IBM Data Cataloging user who has the **dataadmin** role. The username field takes the format of <domain_name>/<username>. To determine a domain and username with the **dataadmin** role, go to the Access Users page in the IBM Data Cataloging GUI and click the view for the defined users.

      For the local domain, it is not necessary to specify the domain as part of the username field as it is the default domain. For example, to define username for user1 in the local domain that is assigned the **dataadmin** role, in the configuration file, enter the following value: **username=user1**

.

```
[datacataloging]
# Name of the Spectrum Scale connection to scan files from
# Check using the Spectrum Discover connection manager APIs
scale_connection=fs3
# Data Cataloging Master Node IP
master_node_ip=203.0.113.23
# Data Cataloging user name, having 'dataadmin' role
# Use format <domain_name>/<username>
# e.g. username=Scale/scaleuser1
username=user1
```

Note: The scanner output file generates approximately 1 K of metadata for every file in the system. If there are 12 M files, the size is expected to be approximately 12 GB. By default, the output file is written to the same directory that is being scanned. The log file output location can be customized by setting the `outputdir` field.

5. Run the scan by using the following command:

```
./scale_scanner.py
```

Note: While you run the **./scale_scanner.py** command, you can start another scan. If you start another scan, ensure that you run the scan with another connection that is online and is not being scanned currently. When the scanner is running, the scanner hides the scan now automatically.
Note: As you run the scale_scanner.py script, you are prompted for the password for the IBM Data Cataloging user that is configured in the scale_scanner.conf file with the username under the `datacataloging` section. You must provide the correct password for the configured user. As described in the configuration file, this user needs to be a valid user configured in the IBM Data Cataloging Authentication service (Access management). Also, this user must be assigned to the `dataadmin` role.
For example:

```
$ ./scale_scanner.py
Enter password for SD user 'user1':
Scale Scan Policy is created at: ./scanScale.policy
```

Note:

- After you see a line similar to "`0 'skipped' files and/or errors`", press enter to return to the command prompt.
- The scan takes approximately 2 minutes 30 seconds for every 10 M files on the following configuration:

```
x86 -based Spectrum Scale Cluster
•4 M4 NSD client nodes
•2 M4 NSD server nodes
•DCS3700 350 2TB NL SAS drives & 20 200GB SSD
•QDR InfiniBand cluster network
```

Copying the output of the IBM Storage Scale file system scan to the IBM Data Cataloging master node

After you have scanned your IBM Spectrum Scale file system and have the list.metaOcean output file, copy it to the IBM Data Cataloging master node.

Procedure

As an IBM Data Cataloging administrator, use the **scp** command to copy the list.metaOcean file from the scan output directory to the /opt/ibm/metaocean/data/producer directory on the master node.
Note: If there are multiple file systems in the same cluster that are being scanned, you can rename the list.metaOcean file to avoid name conflicts and to not overwrite an existing list.metaOcean file that is in use. For example:

```
$ mv list.metaOcean list.metaOcean.myfilesystem
```

```
$ scp list.metaOcean.myfilesystem moadmin@MasterNodeIP:/opt/ibm/metaocean/data/producer
```

Ingesting metadata from IBM Storage Scale file system scan in IBM Data Cataloging

Records are inserted into IBM Data Cataloging for indexing when they are pushed to a Kafka connector topic corresponding to the type of data being ingested. In the case of IBM Storage Scale, the Kafka connector topic type is `scale-scan-connector-topic`.

About this task

A Kafka client producer is required to put the IBM Storage Scale file system scan file records onto the Kafka connector topic. The following steps show how to use the **ingest** alias command to push the records in the list.metaOcean file (or another named file) onto the Kafka connector topic.

Procedure

1. Run the following command to ingest the data:

```
$ ingest /opt/ibm/metaocean/data/producer/list.metaOcean
```

2. Replace the list.metaOcean path with the path of the file that you want to ingest.

Ingesting quota information from the file system

The file system scanning tool, IBM Storage Scale Scanner, has the ability to harvest and send quota information to IBM Data Cataloging.

Procedure

To perform quota ingestion, run the following command on the IBM Storage Scale cluster node:

```
./scale_scanner.py --quota-only
```

For example:

```
$ sudo ./scale_scanner.py --quota-only
Enter password for SD user 'user1':
```

IBM Storage Archive data source connections

You can define tags and policies in IBM Data Cataloging based on values that are derived from IBM Spectrum® Archive metadata to help in searching and categorizing files.

IBM Data Cataloging integrates with IBM Storage Archive to display search results that include the following archive state of files:

Migration status `migstatus`
: Search results display details for the following migration status:

  `migrtd`
  : Indicates that the file is migrated to tape.

  `resdnt`
  : Indicates that the file is resident in the file system.

  `premig`
  : Indicates that the file is pre-migrated to tape.

Migration "location" `migloc`
: Search results display information on the tape cartridge in the following format: "1 *tape cartridge volser@tape storage pool id@tape library serial number*". Any additional copies must be separated by colons.

Actual size of file in the associated IBM Storage Scale file system `Size Consumed Bytes`
: IBM Data Cataloging displays a zero if the file is moved to tape.

Remember: The migration state information is collected and summarized in the IBM Data Cataloging State facet. You can access this facet by using IBM Data Cataloging visual search. For more information, see Searching.For more information, see the topic *Searching* in the *Data Cataloging: Administration Guide*.
The IBM Data Cataloging interface displays the search results including the metadata information.

Important: The location information that is displayed in IBM Data Cataloging is provided by IBM Storage Scale. It corresponds to the `dmapi.IBMTPS` attribute for the file. Run the **mmlsattr -L** command for more details.

IBM Cloud Object Storage data source connection

You can create the IBM Cloud® Object Storage (COS) connection and initiate a scan.

IBM® COS uses a connector residing on the storage system to push events to a Kafka topic residing in the IBM Data Cataloging cluster. When configured, the IBM Data Cataloging consumes the events and indexes them into the IBM Data Cataloging database.
Restriction: IBM Data Cataloging uses a unit separator (ASCII code 0x1F) as the field delimiter for ingestion into the database. This means that data which contains this character in path/file/object names results in improper parsing of the input data and the records are rejected by IBM Data Cataloging.

- **Prerequisites**
  The IBM Cloud Object Storage Scanner and Replay prerequisites are listed:
- **Creating an IBM Cloud Object Storage data source connection**
  You can create an IBM Cloud Object Storage (COS) data source connection and from the storage system.
- **Scanning an IBM Cloud Object Storage data connection**
  You can initiate an IBM connection scan to collect system metadata from an IBM Cloud Object Storage system.
- **Best practices for scanning IBM Cloud Object Storage systems**
  Use best practices for scanning IBM Cloud Object Storage (IBM COS) systems.
- **Enabling bucket notifications for Ceph Object Storage**
  Use this information to enable bucket notifications for Ceph® Object Storage.
- **Enabling bucket notifications for Ceph Object Storage on Red Hat OpenShift**
  Use this information to enable bucket notifications for Ceph Object Storage on Red Hat® OpenShift®.
- **Replaying IBM Cloud Object Storage notifications**
  Use the IBM Cloud Object Storage (COS) Replay feature to resend notifications that failed because of an outage or loss of data.
- **Configure IBM Cloud Object Storage notifications for IBM Data Cataloging**
  Ingesting IBM Cloud Object Storage event records into IBM Data Cataloging requires the user to enable the Notification service on the IBM Cloud Object Storage system. Thereafter, the user must connect the IBM Cloud Object Storage system to the IBM Cloud Object Storage connector Kafka topic on the IBM Data Cataloging cluster. The name of this connector topic is `cos-le-connector-topic`.
- **Enabling IBM Cloud Object Storage notification services**
  The IBM Cloud Object Storage notification service can be enabled with the information that follows:
- **Testing the IBM Cloud Object Storage notification service**
  To test the IBM Cloud Object Storage notification service, the tester can populate the IBM Cloud Object Storage vault with test data.

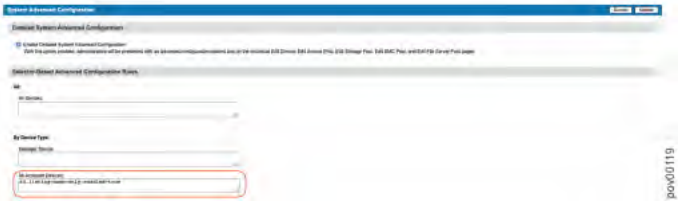
Prerequisites

The IBM Cloud® Object Storage Scanner and Replay prerequisites are listed:

IBM Cloud Object Storage Scanner prerequisite
: For the Scanner, you must enable the Get Bucket Extension for all accesser devices.
To enable the Get Bucket Extension, you must set the `s3.listing-name-only-enabled` equal to true in the Manager System Advanced Configuration.

See Figure 1.

Figure 1. Example of the system advanced configuration



Remember: You do not need to restart the Accesser, but you might need to wait for 5 minutes before the setting takes effect if you do not restart it.

IBM Cloud Object Storage Replay prerequisite
: For the Replay, **`access_logs`** are uploaded to the management vaults within 1 hour after rotation. Rotation can be triggered earlier by setting the Rotation Period to the minimum value of 15 minutes in Manager under Maintenance/Logs/Device Log Configuration. Refer to the IBM Cloud Object Storage System documentation to make sure that this is configured, and the relevant access logs are present before you run the Replay.

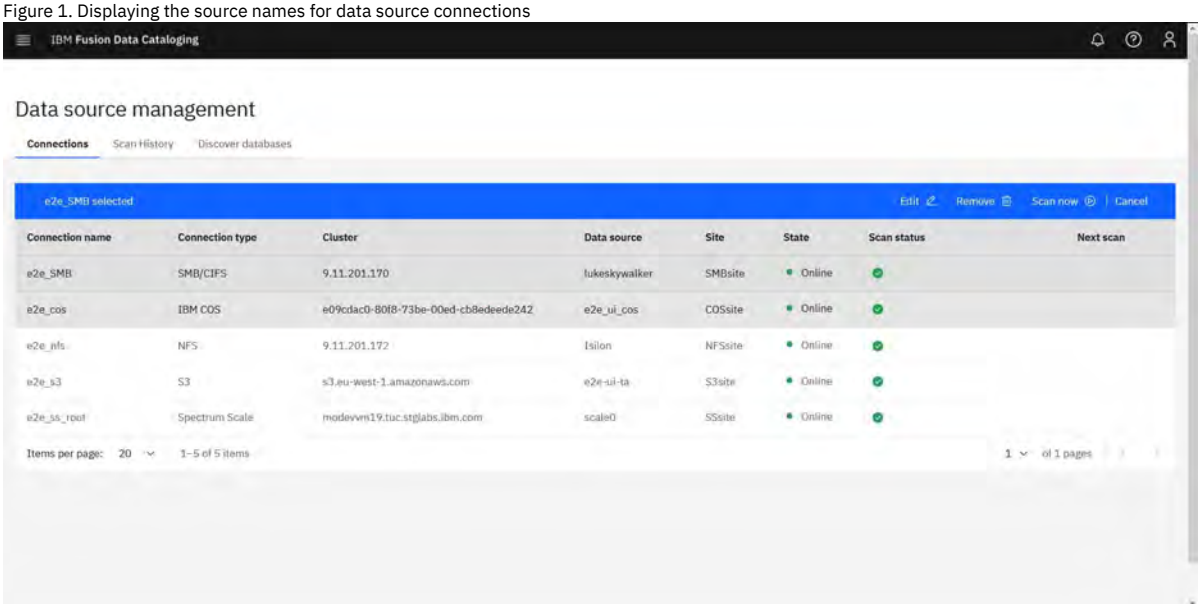
Creating an IBM Cloud Object Storage data source connection

You can create an IBM Cloud® Object Storage (COS) data source connection and from the storage system.

Procedure

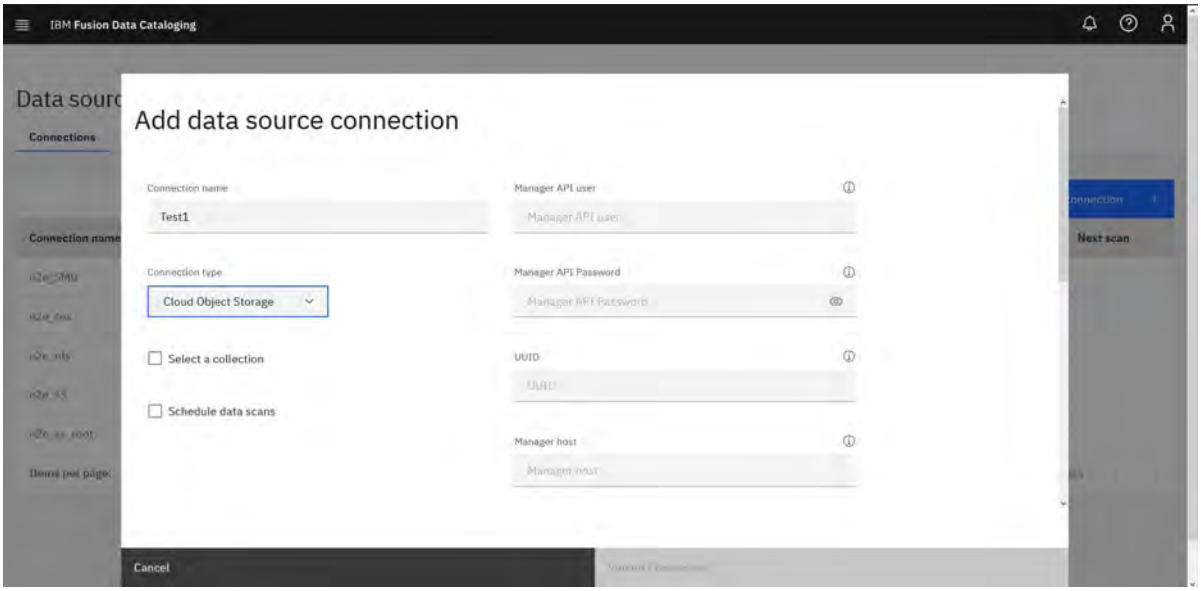
1. Log in to the IBM Data Cataloging web interface with a user ID that has the Data Admin role that is associated with it.
   The data admin access role is required for creating connections. For more information, see Role-based access control.
2. Click menu and click Data connections.
   Clicking Connections displays the different types of data source connection names, connection type, clusters, data source, site, state, scan status, next scan , and Add Connection.

   Figure 1. Displaying the source names for data source connections



3. Click Add connection to display a new window that shows Add data source connection.

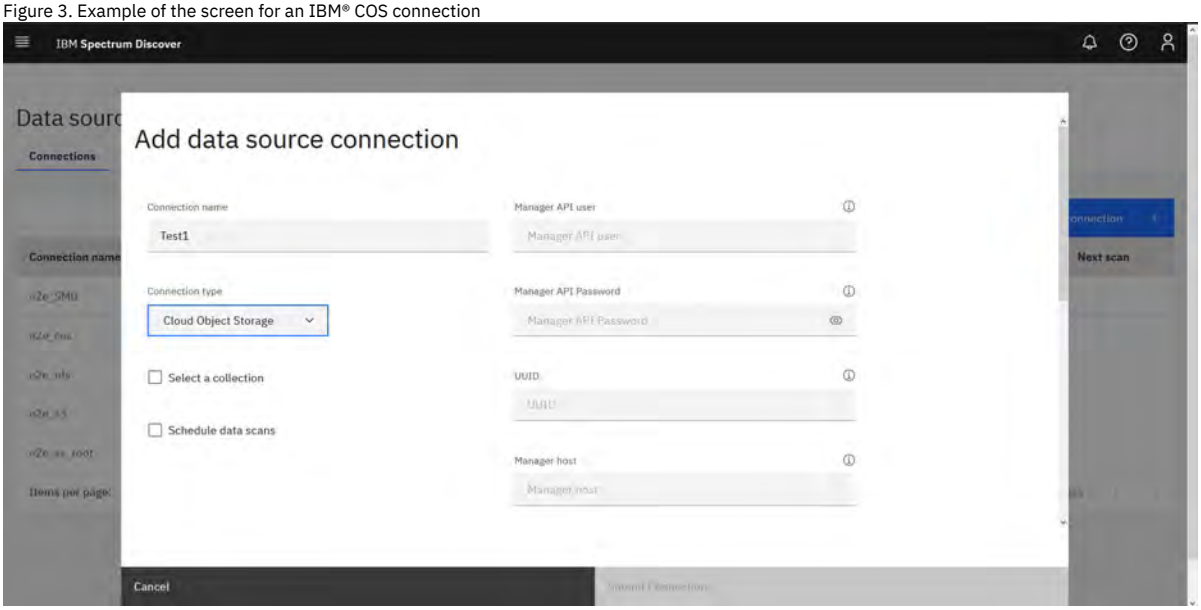
   Figure 2. Example of window that shows Data Connections Add data source Connection

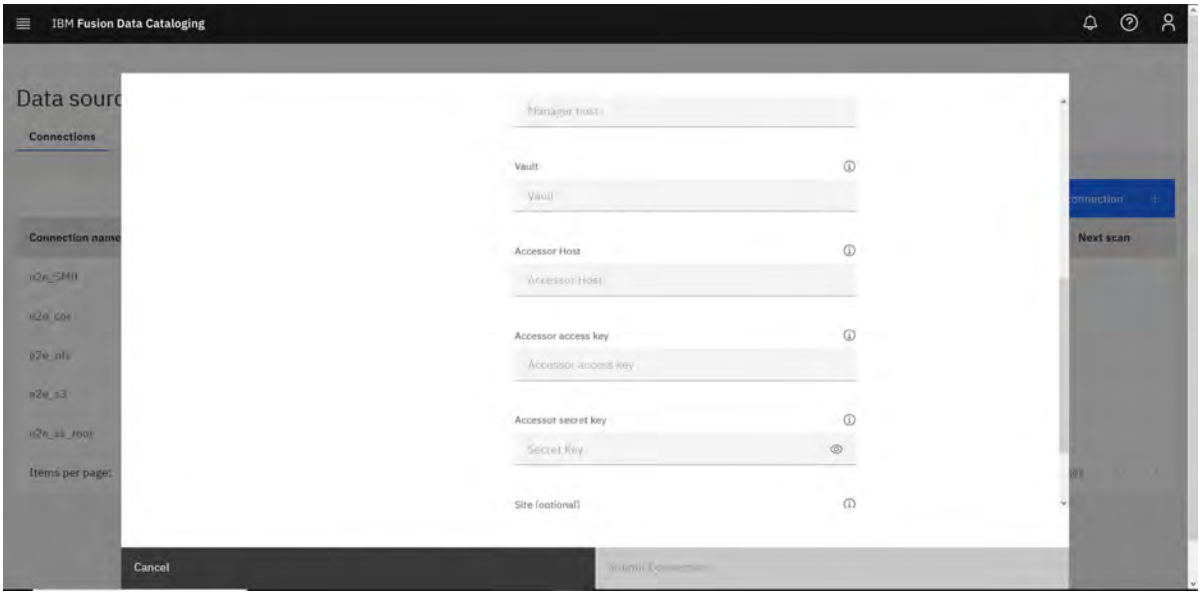


4. Do the following steps:
    a. In the field for Connection name, define a Connection name.
    b. Choose the type of data source connection from the Connection type list.
5. Select the connection type Cloud Object Storage.

[Figure 3]

Figure 3. Example of the screen for an IBM® COS connection





6. In the screen for Cloud Object Storage, complete fields, and click Submit Connection.
   For Cloud Storage Object Connections Manager

   Manager API user
   : A user ID that has permissions to connect to the data source system.

   Manager API Password
   : The password for the user ID specified in the previous field.

   UUID
   : The unique ID of the DSNet cluster. To obtain the UUID, log in to the COS Manager GUI and click Help > About this system on the upper-right corner of the window.

   Host
   : The IP or hostname of the manager node within the DSNet.

   Vault
   : The specific data vault represented by this connection.

   Site
   : An optional physical location tag that an administrator can provide to see the physical distribution of their data.

   Accesser
   : The IP address or hostname of the Accesser® node on DSNet.

   Accesser access key
   : The Accesser access key that has permission to access data in the data vault that is to be scanned. If the accessor access key value is blank, the value is retrieved (for the manager API user) from the manager API.

   Accesser secret key
   : The Accesser secret key that has permission to access data in the vault that is to be scanned. If the secret access key value is blank, the value is retrieved (for the manager API user) from the manager API.

Scanning an IBM Cloud Object Storage data connection

You can initiate an IBM® connection scan to collect system metadata from an IBM Cloud® Object Storage system.

About this task

When you initiate a scan from the IBM Data Cataloging graphical user interface (GUI), the metadata is transferred asynchronously back to the IBM Data Cataloging instance.

Note:
IBM Data Cataloging does not support scanning of vaults in a dsNet that has any of the following things:

- Proxy vault
- Mirrored vault
- Vault setup for migration

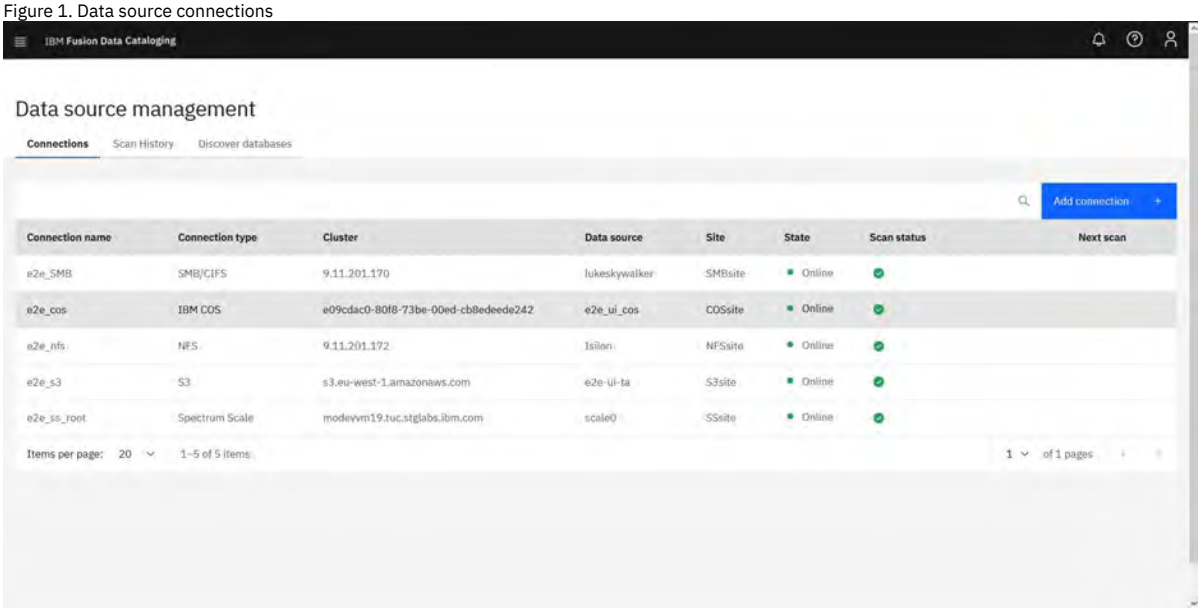
Automated scanning and data ingestion relies on an established and active network connection between the IBM Data Cataloging instance and the IBM Cloud Object Storage storage source. If the connection cannot be established, the state of the data source connection shows as unavailable, and the option for automated scanning does not appear in the IBM Data Cataloging GUI for that connection.

Note: If a scan does not complete successfully, check the log file for errors and warnings. If the error or warning message indicates a need to check the configurations file or the settings file, then you must modify the file as required. For example, in some cases you must update the override warnings in the settings file by adding: `"override_warnings": true` at the root level.
The settings or the configuration file is available in the following location: /opt/ibm/metaocean/data/connections/cos/scan/scanner-settings.json

Procedure

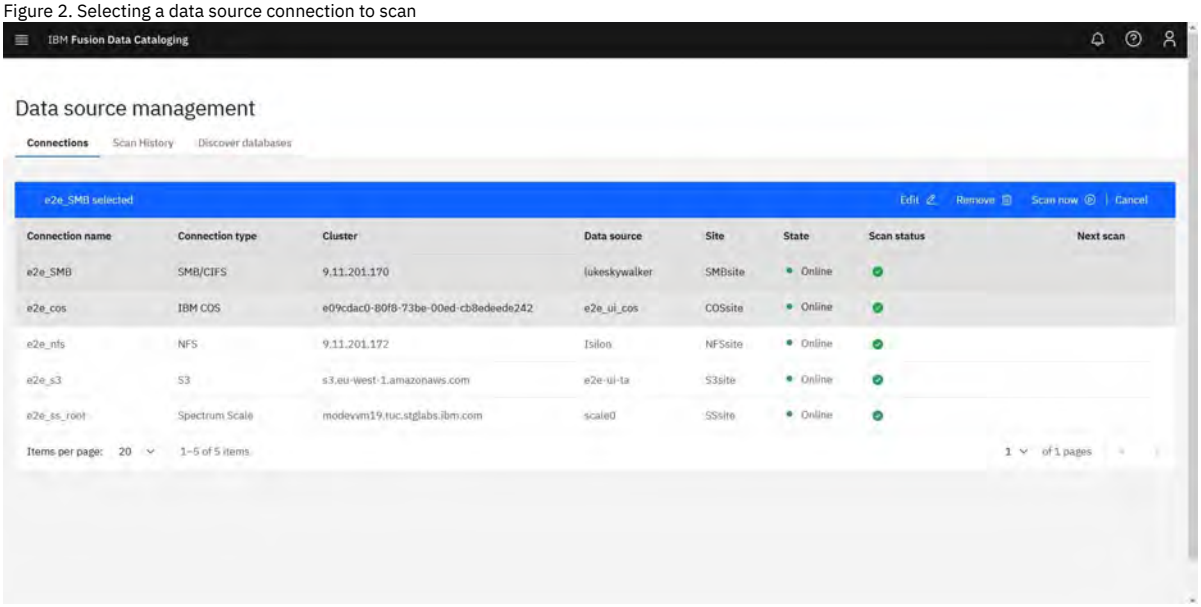
1. Log in to the IBM Data Cataloging graphical user interface (GUI).
2. Click menu and go to Data connections > Connections.
   The following example shows the Data connections menu page:

Figure 1. Data source connections



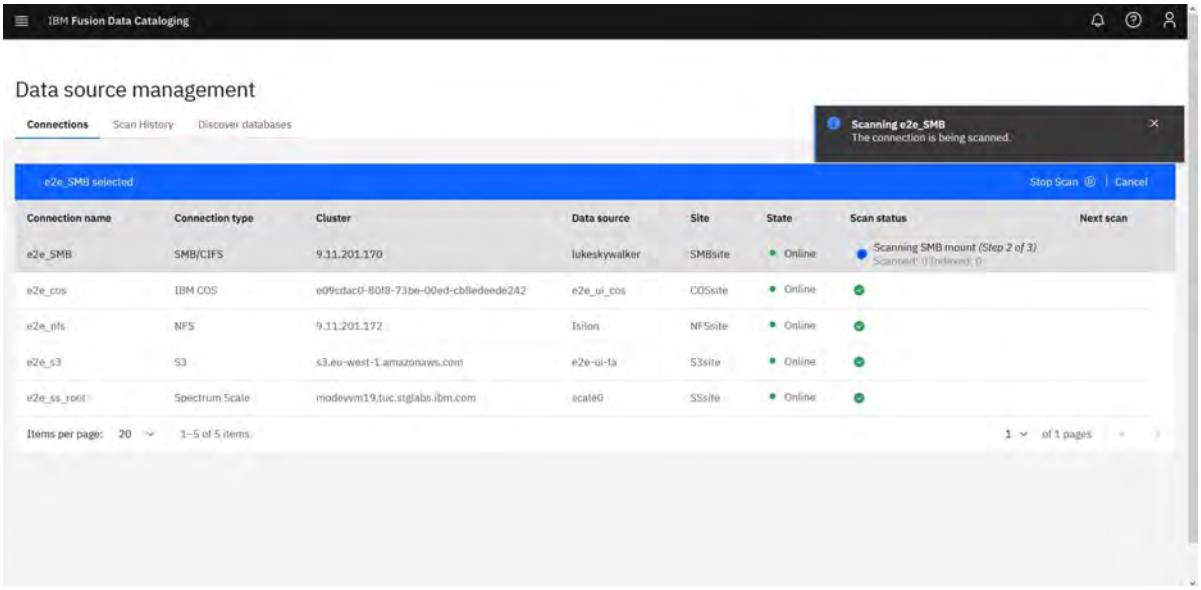
3. Select the data source connection name that you want to scan. Ensure that the **`State`** is listed as **`Online`** to make your system scan ready.
   The following example shows how to connect to the IBM Data Cataloging library.

Figure 2. Selecting a data source connection to scan



4. Select Scan now to change the status to Scanning.
   The following example shows an active scan.

Figure 3. Active scans



5. When the scan finishes, the state field returns to a status of Online.

Best practices for scanning IBM Cloud Object Storage systems

Use best practices for scanning IBM Cloud® Object Storage (IBM® COS) systems.

It is recommended to check the log files in the following directories after each scan:

/opt/ibm/metaocean/data/connections/cos/<connection_name>/debug/<scan_timestamp>/scanner.debug indicates whether the scan was successful or not.

/opt/ibm/metaocean/data/connections/cos/<connection_name>/error/<scan_timestamp>/scanner.error contains a list of all the messages that are not delivered to IBM Data Cataloging.

This file contains a list of all the messages that are not delivered to IBM Data Cataloging.

/opt/ibm/metaocean/data/connections/cos/<connection_name>/data/<scan_timestamp>/ contains a subfolder with the scanned data source name. There is a stats folder inside this folder that contains information about the number of objects in the data source or the number of objects or scanned files.

You can also compare the total size of the bucket that is reported in IBM Data Cataloging with the total size of the IBM COS at its source (if it is available).

Enabling bucket notifications for Ceph Object Storage

Use this information to enable bucket notifications for Ceph® Object Storage.

Before you begin

Ensure you have the following features set up:

- IBM Data Cataloging 2.0.2.1
- Red Hat® Ceph Storage 4.0 (available starting with version Beta 8)
- A Ceph Object Gateway node that is set up with an HTTPS endpoint

Restriction:

- Ceph Object bucket names must be unique across all data sources. You cannot use the same bucket name to reach a Ceph data source. For example, if there is the IBM Cloud® Object Storage or Amazon S3 bucket with the name `"my_bucket"`, you cannot reach a Ceph data source with the bucket name `"my_bucket"`.
- Notifications from versioned buckets are not supported.
- Only one IBM Data Cataloging node can be configured for push notifications from Ceph Object Storage cluster at a time.

About this task

Use the following steps to enable bucket notifications for Ceph Object Storage.

Procedure

1. Create a data source connection to the Ceph Object Storage cluster.
   A Ceph Object Storage source is established as an Amazon S3 data source connection.
   Remember: Each bucket must have its own data source connection entry in IBM Data Cataloging.
2. To enable Ceph Object Storage bucket notifications:
   a. Copy the ca.crt file from IBM Data Cataloging node to a directory on the Ceph Object Gateway nodes.

b. Locate the file in the /etc/kafka directory on the IBM Data Cataloging node.
c. Give this file a unique name on the Ceph node after it is copied over.
Remember: Ensure that the file has the same name and in the same location on each Ceph Object Gateway node.
You can choose to use /etc/ssl/certs as the copy target directory on the Ceph Object Gateway node.

3. Create a topic entity by using Ceph bucket notification REST API. The topic contains the push endpoint on IBM Data Cataloging where the notifications are sent to.
Remember: To enable notifications to be sent to IBM Data Cataloging you must provide push endpoint parameters when you create the topic entity.
These parameters include the IBM Data Cataloging Kafka topic and credentials that are required to securely produce messages to the topic. For more information about the REST API, see Create a Topic*Create a Topic* in the Ceph documentation.
The following parameters must be in the POST request:

```
POST
Action=CreateTopic
&Name=ceph-le-connector-topic
&push-endpoint=<endpoint>
&Attributes.entry.5.key=use-ssl&Attributes.entry.5.value=true
&Attributes.entry.6.key=ca-location&Attributes.entry.6.value=<file path>
```

In this example:

<endpoint>
Indicates the URI of the IBM Data Cataloging Kafka broker in this format: kafka://cos:<password>@<discover_fqdn>:9092
<password>
Indicates the password that can be obtained by an administrator on the IBM Data Cataloging node from the following location: /etc/kafka/sasl_password
<discover_fqdn>
Indicates the fully qualified domain name of the IBM Data Cataloging node.
<file path>
Indicates the location and file name of the Kafka certificate authority (CA) file on the Ceph Object Gateway Node.

The following example shows topic creation by using the s3curl utility:

```
$ ./s3curl.pl --id=rhceph -- -k -X POST https://<ceph object gateway address>:8080/ -d
"Action=CreateTopic&Name=ceph-le-connector-topic&push-endpoint=kafka://cos:
<password>@<discover_fqdn>:9092&Attributes.entry.5.key=use-ssl&Attributes.entry.5.value=true&
Attributes.entry.6.key=ca-location&Attributes.entry.6.value=/etc/ssl/certs/ca.crt"
```

The --id parameter identifies the credentials to use in the **s3curl** configuration file.

4. Create a notification entity by using the Ceph bucket REST API. This associates events on a specific bucket to a topic. For more information, see CREATE NOTIFICATION *CREATE NOTIFICATION* in the Ceph documentation.
The following example shows how to establish a bucket notification by using the s3curl utility:

```
$ ./s3curl.pl --id=rhceph --put=notif.xml -- -k https://<ceph object gateway address>:8080/<bucket>?notification


Contents of notif.xml:
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2010-03-31/">
    <TopicConfiguration>
        <Id>id1</Id>
        <Topic>arn:aws:sns:default::ceph-le-connector-topic</Topic>
    </TopicConfiguration>
</NotificationConfiguration>
```

You can now capture events on objects within the configured buckets.

Enabling bucket notifications for Ceph Object Storage on Red Hat OpenShift

Use this information to enable bucket notifications for Ceph® Object Storage on Red Hat® OpenShift®.

Before you begin

Ensure you have the following features set up:

- IBM Data Cataloging 2.0.4
- Red Hat Ceph Storage 4.1z2 or later
- A Ceph Object Gateway node that is set up with an HTTPS endpoint

Restriction:

- Ceph Object bucket names must be unique across all data sources. You cannot use the same bucket name to reach a Ceph data source. For example, if an IBM Cloud® Object Storage or Amazon S3 bucket exists with the name **"my_bucket"**, you cannot reach a Ceph data source with the bucket name **"my_bucket"**.
- Notifications from versioned buckets are not supported.
- Only one IBM Data Cataloging node can be configured for push notifications from Ceph Object Storage cluster at a time.

About this task

Use the following steps to enable bucket notifications for Ceph Object Storage.

Procedure

1. Create a data source connection to the Ceph Object Storage cluster.
A Ceph Object Storage source is established as an Amazon S3 data source connection.

Remember: Each bucket must have its own data source connection entry in IBM Data Cataloging.

2. To enable Ceph Object Storage bucket notifications:
   a. Issue the following command to transfer the ca.crt file from IBM Data Cataloging node to the Ceph Object Gateway nodes.

      ```
      oc get secret kafka -n ibm-data-cataloging -o jsonpath='{.data.sasl_ca\.crt}' | base64 -d > ca.crt
      ```

   b. Give this file a unique name on the Ceph node after it is copied over.
      Remember: Ensure that the file has the same name and in the same location on each Ceph Object Gateway node.
      You can choose to use /etc/ssl/certs as the copy target directory on the Ceph Object Gateway node.
3. Create a topic entity by using Ceph bucket notification REST API. The topic contains the push endpoint on IBM Data Cataloging where the notifications are sent to.
   Remember: To enable notifications to be sent to IBM Data Cataloging, you must provide endpoint parameters when you create the topic entity.
   These parameters include the IBM Data Cataloging Kafka topic and credentials that are important to securely produce messages to the topic. For more information about the REST API, see https://docs.ceph.com/docs/master/radosgw/notifications/#create-a-topic.
   The following parameters must be in the POST request:

   ```
   POST
   Action=CreateTopic
   &Name=ceph-le-connector-topic
   &push-endpoint=<endpoint>
   &Attributes.entry.5.key=use-ssl&Attributes.entry.5.value=true
   &Attributes.entry.6.key=ca-location&Attributes.entry.6.value=<file path>
   ```

   The parameters that are shown in the example are explained in the following section.

   <endpoint>
   Indicates the URI of the IBM Data Cataloging Kafka broker in this format: kafka://cos:<password>@<discover_fqdn>:443
   <password>
   Indicates the password that can be obtained by an administrator on the Red Hat OpenShift node. Issue the following command to obtain the password:

   ```
   oc get secret kafka-sasl-password -n ibm-data-cataloging -o jsonpath='{.data.password}'
   ```

   <discover_fqdn>
   Indicates the fully qualified domain name of the IBM Data Cataloging node.
   <file path>
   Indicates the location and file name of the Kafka certificate authority (CA) file on the Ceph Object Gateway Node.

   The following example shows topic creation by using the s3curl utility:

   ```
   $ ./s3curl.pl --id=rhceph -- -k -X POST https://<ceph object gateway address>:8080/ -d
   "Action=CreateTopic&Name=ceph-le-connector-topic&push-endpoint=kafka://cos:
   <password>@<discover_fqdn>:9092&Attributes.entry.5.key=use-ssl&Attributes.entry.5.value=true&
   Attributes.entry.6.key=ca-location&Attributes.entry.6.value=/etc/ssl/certs/ca.crt"
   ```

   The --id parameter identifies the credentials to use in the **s3curl** configuration file.

4. Create a notification entity by using the Ceph bucket REST API. This associates events on a specific bucket to a topic. For more information, see:
   https://docs.ceph.com/docs/master/radosgw/s3/bucketops/#create-notification
   The following example shows how to establish a bucket notification by using the s3curl utility:

   ```
   $ ./s3curl.pl --id=rhceph --put=notif.xml -- -k https://<ceph object gateway address>:8080/<bucket>?notification


   Contents of notif.xml:
   <NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2010-03-31/">
       <TopicConfiguration>
           <Id>id1</Id>
           <Topic>arn:aws:sns:default::ceph-le-connector-topic</Topic>
       </TopicConfiguration>
   </NotificationConfiguration>
   ```

   You can now capture events on objects within the configured buckets.

Replaying IBM Cloud Object Storage notifications

Use the IBM Cloud® Object Storage (COS) Replay feature to resend notifications that failed because of an outage or loss of data.

The IBM® COS Replay reads object metadata from vaults and submits the metadata to IBM Data Cataloging by using Kafka notifications.

- **Overview of architecture**
  This topic describes a high-level overview of IBM Cloud Object Storage Scanner architecture.
- **Configuration file**
  The configuration file is used by the Notifier and Replay.
- **Replay performance**
  The number of requests that are issued by IBM Cloud Object Storage Replay is throttled to ensure that overall dsNet performance remains at the agreed level.
- **Replay tasks and vault settings**
  A few scenarios exist that prevent the Replay from operating correctly.
- **Including and excluding vaults**
  You can set the vaults that you scan with various settings in the configuration file.
- **Stats files**
  The IBM Cloud Object Storage Scanner tracks each LIST process status to a stats file.
- **Replay**
  When a severe outage occurs and causes the loss of notifications sent by the system to IBM Data Cataloging, the IBM Cloud Object Storage Scanner Replay feature can be used to recover lost notifications.

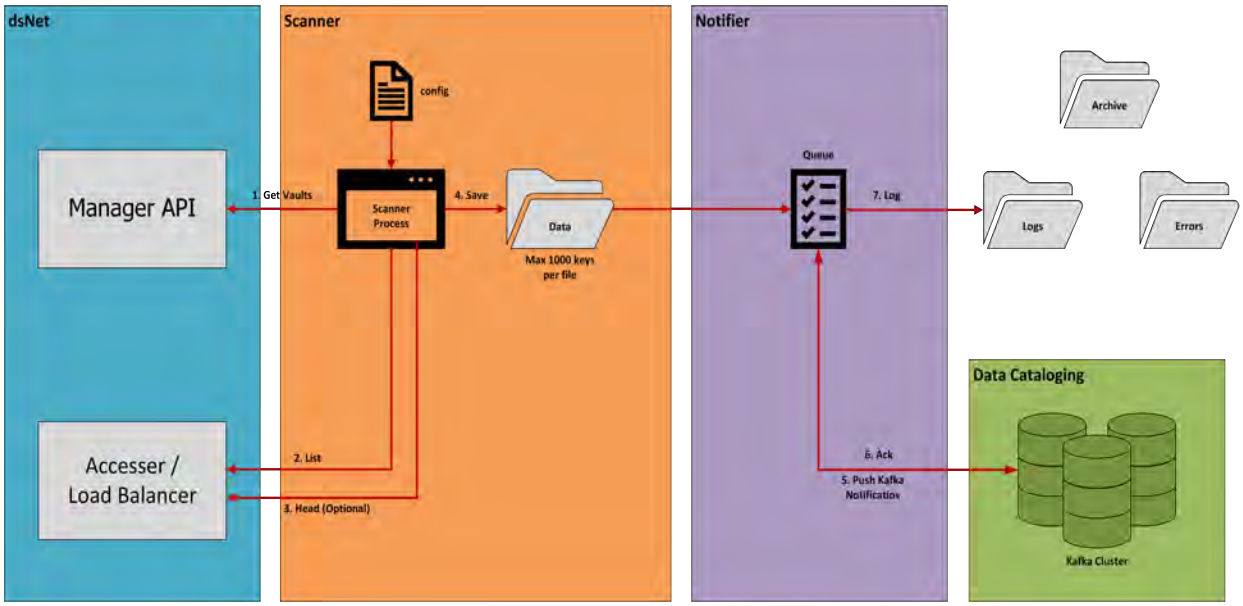
- **Initialization for Replay**
  During the startup, Replay reads the configuration file and issues requests to the Manager of the dsNet device similar to the Scanner.
- **Error conditions**
  Sometimes Replay does not have enough information to replay the original notification. If this occurs, you must fix the problems manually.
- **Output**
  Messages are batched by 1,000 or to the Scanner list objects size configuration setting, if specified.
- **Renaming a vault for Replay**
  When you rename a vault, it is possible that Replay can abort.
- **Starting the Replay**
  The guidelines and rules for using Replay are documented in this topic.
- **Debug mode for Replay**
  Run Replay in debug mode to troubleshoot problems.
- **Notifier**
  The Notifier is the component that reads the JSON notifications that are written by Scanner or Replay and sends notifications to the Kafka cluster.
- **Limitations**
  Limitations apply when the Notifier uses a Kafka configuration retrieved from the Manager API.
- **Starting the Notifier**
  Running the Notifier has rules and limitations.
- **Notifier operation**
  The Notifier enumerates and processes all .log files in the Scanner data directory.
- **Stopping the Notifier**
  You might need to stop and restart the Notifier.
- **Restarting the Notifier**
  When you stop the Notifier following a shutdown with the kill.notifier file, you must rename the file manually or delete the file before you do a restart.
- **Progress report**
  The Progress Report provides an instant snapshot of status for the Scanner and Notifier.
- **Logging**
  You can view the list of directories that are generated by scanner, notifier, and replay.
- **IBM Cloud Object Storage Scanner output data**
  The Scanner generates a directory beneath the output data directory for each vault or vault prefix as defined in the configuration file.
- **Appendix**
  The appendix shows an example of a log file and examples of Scanner debug data.

Overview of architecture

This topic describes a high-level overview of IBM Cloud® Object Storage Scanner architecture.

The following figure shows a high-level overview of IBM Cloud Object Storage Replay architecture.

Figure 1. IBM Cloud Object Storage Replay architecture



The /opt/ibm/metaocean/data/connections/cos/replay/output/data folder puts the replay output and the notifier reads Kafka messages from the directory. Putting replay output onto disk means that a cold restart is possible.

The IBM Cloud Object Storage Replay consists of two major components:

Replay
: Downloads system's logs and re-creates notifications that are sent during a defined time period.

Notifier
: Submits the extracted information to IBM Data Cataloging.

Configuration file

The configuration file is used by the Notifier and Replay.

The configuration file includes:

- Information regarding the net
- Runtime parameters for the Notifier and Replay
- A list of vaults to scan

The configuration file is named scanner-settings.json and must sit in the /opt/ibm/metaocean/data/connections/cos/replay directory.

The rules for IBM Cloud® Object Storage Replay settings are:

- All access logs are scanned.
- All objects that are created or updated since Coordinated Universal Time (UTC) 00:00:01 from 11 April 2018 to Coordinated Universal Time (UTC) 10:01:53 on 21 September 2018 are scanned in batches of 1000.
- Custom metadata is retrieved for each object or version.
- Ten vaults are processed in parallel.
- Each vault has a single process LIST that issues requests and 15 processes that issue HEAD requests.

The following example shows every setting. Most settings have default values and can be omitted, but these screens show a typical example by using default values.

Example of the Cloud Object Storage Replay settings

```
{
     "system": {
     "name": "Test dsnet",
     "uuid": "00000000-0000-0000-0000-000000000000",
     "manager_ip": "172.1.1.1",
     "accesser_ip": "172.1.1.2",
     "accesser_supports_https": false,
     "manager_username": "admin",
     "manager_password": "password",
     "is_ibm_cos": true
},
   "timestamps": {
     "min_utc": "2018-01-01T00:00:00Z",
     "max_utc": "2018-09-21T10:01:53Z"
},
   "policy_engine" : {
     "spectrum_discover_host": "modevvm32.tuc.stglabs.ibm.com"
     "user": "sdadmin",
     "password": "password"
},
   "scanner": {
     "max_requests_per_second": 5000,
     "max_parallel_list": 10,
     "parallel_head_per_list": 5,
     "list_objects_size": 100
},
"notifier":{
     "kafka_format": 1,
     "kafka_endpoint": "192.168.1.1:9092",
     "kafka_topic": "cos-le-connector-topic",
     "kafka_username": "cos",
     "kafka_password": "password",
     "kafka_pem": "-----BEGIN CERTIFICATE-----...\n-----END CERTIFICATE-----\n"
},
   "logging": {
     "debug_log_max_bytes": 10000000,
     "debug_log_backup_count": 10000,
     "notification_log_max_bytes": 10000000,
     "notification_log_backup_count": 10000,
     "notification_log_all": true
},
   "include_all_vaults": false,
   "has_custom_metadata": true,
   "override_warnings": true,
   "exclude-vaults": ["Manager"],
   "vaults": [
     {
       "vault_name": "Vault-1"
     },
     {
       "vault_name": "Vault-2",
       "has_custom_metadata": false
     },
     {
       "vault_name": "Vault-3",
       "has_custom_metadata": false,
       "prefix": "customers/live"
     }
  ]
}
```

Typical Cloud Object Storage configuration settings

```
{
```

```
    "dsnet": {
    "name": "Test dsnet",
    "uuid": "00000000-0000-0000-0000-000000000000",
    "manager_ip": "172.1.1.1",
    "accesser_ip": "172.1.1.2",
    "accesser_supports_https": false,
    "manager_username": "admin",
    "manager_password": "password",
    "is_ibm_cos": true
},
  "timestamps": {
    "min_utc": "2018-01-01T00:00:00Z",
    "max_utc": "2018-09-21T10:01:53Z"
},
  "policy_engine" : {
    "spectrum_discover_host": "modevvm32.tuc.stglabs.ibm.com"
    "user": "sdadmin",
    "password": "password"
},
  "scanner": {
    "max_requests_per_second": 5000,
    "max_parallel_list": 10,
    "parallel_head_per_list": 5,
    "list_objects_size": 100
},
  "notifier":{
    "kafka_format": 1,
    "kafka_endpoint": "192.168.1.1:9092",
    "kafka_topic": "cos-le-connector-topic",
    "kafka_username": "cos",
    "kafka_password": "password",
    "kafka_pem": "-----BEGIN CERTIFICATE-----...\n-----END CERTIFICATE-----\n"
},
  "logging": {
    "debug_log_max_bytes": 10000000,
    "debug_log_backup_count": 10000,
    "notification_log_max_bytes": 10000000,
    "notification_log_backup_count": 10000,
    "notification_log_all": true
},
  "include_all_vaults": false,
  "has_custom_metadata": true,
  "override_warnings": true,
  "exclude-vaults": ["Manager"],
  "vaults": [
    {
      "vault_name": "Vault-1"
    },
    {
      "vault_name": "Vault-2",
      "has_custom_metadata": false
    },
    {
      "vault_name": "Vault-3",
      "has_custom_metadata": false,
      "prefix": "customers/live"
    }
  ]
}


{
    "dsnet": {
    "manager_ip": "192.168.2.106",
    "accesser_ip": "192.168.2.111"
},
    "timestamps": {
    "min_utc": "2018-04-11T00:00:01.000Z",
    "max_utc": "2018-09-21T10:01:53Z"
},
    "scanner":{
       "max_requests_per_second": 5000
},
    "include_all_vaults": true
}

{
    "system": {
    "manager_ip": "192.168.2.106",
    "accesser_ip": "192.168.2.111"
},
   "policy_engine" : {
    "spectrum_discover_host": "modevvm32.tuc.stglabs.ibm.com"
},
    "timestamps": {
    "min_utc": "2018-04-11T00:00:01.000Z",
    "max_utc": "2018-09-21T10:01:53Z"
},
    "scanner":{
       "max_requests_per_second": 5000
},
    "include_all_vaults": true
}
```

IBM Cloud Object Storage Scanner is highly configurable. Each element in the file is described in Table 1.

Remember: IBM Data Cataloging does not support file or file path names that use characters that are not part of the UTF-8 character set.

Table 1. Explanation of the configuration file

Element	Description	Optional	Default value	Restart scanner if changed	Restart notifier if changed
System section					
name	Free-text name of the dsNet. Appears in the 'system_name' element in all Kafka messages.	✓	Retrieved from Manager API if configured. If not, the name does not appear in Kafka messages.	✓	✗
uuid	UUID of the dsNet. Appears in the 'system_uuid' element in all Kafka messages.	✓	Retrieved from Manager API.	✓	✗
manager_ip	Single IP address or host name of the manager device.	✗	Not applicable	✓	✗
accesser_ip	Single IP address or host name of an accesser device or load balancer that routes to the accessers.	✗	Not applicable	✓	✗
accesser_supports_https	Boolean value that indicates whether http or https can be used when you send requests to the accesser or load balancer.	✓	true	✓	✗
manager_username	Username for accessing the manager API. For testing only. Not to be used in production.	✓	Supplied by user at prompt	✓	✗
manager_password	Password for accessing the Manager API. For testing only. Not to be used in production.	✓	Supplied by user at prompt	✓	✗
is_ibm_cos	Boolean value that indicates whether the system is an IBM Cloud Object Storage or another s3 compliant system. If true, the IBM® Get Bucket Extension is used to retrieve object keys from the vaults. Note: Setting the value to false is not currently supported by the Scanner and Notifier.	✓	True	✓	✗
accesser_access_key	Access key ID for S3 calls to the accesses or load balancer. For testing only. Not to be used in production.	✓	Supplied by user at prompt if you cannot retrieve it from Manager API for the user account that is specified in dsNet/manager_ username.	✓	✗
accesser_secret_key	Secret key for S3 calls to the accesser or load balancer. For testing only. Not to be used in production.	✓	Supplied by user at prompt if you cannot retrieve from Manager API.	✓	✗
Time stamps section					
min_utc	Only objects or version in the vaults that have a **LastModified** datetime on or after this timestamp is submitted to IBM Data Cataloging. Needs to be less than the **max_utc** value. Note: Changing **min_utc** and restarting scanner applies only to objects not yet scanned. Objects scanned before restart might have a **LastModifiedDate** value that is earlier than the **min_utc** value.	✗		✓ See note.	✗
max_utc	Only objects or version in the vaults that have a **LastModified** datetime on or before this time stamp is submitted to IBM Data Cataloging. Needs to be more than min_utc and less than current time. Note: Changing **max_utc** to a more recent time and restarting does not mean that new objects written since the old **max_utc** is scanned. The scanner continues from the last object's key that is scanned in lexicographic order. This means that new objects with names smaller than the last object scanned are not scanned.	✓		✓ See note.	✗
Policy engine section		(Only required for IBM Data Cataloging 2.0.0.3 and later)			
spectrum_discover_host	Host name or IP address of the policy engine service from which the Kafka certificate is retrieved.	✗	none	✓	✓
user	Username for authorization on policy engine.	✗	none	✓	✓
password	Password for authorization on policy engine.	✗	none	✓	✓
Replay section					
access_log_directory	The access_log_directory is where the dsNet access log files are stored after download. Access logs must be in the root input folder. Files in subdirectories are not processed.	✓	[IBM Cloud Object Storage Replay]/ access_logs	Restart Replay if changed	Restart Replay if changed
download	If download is set to false, access logs are not downloaded and are assumed to already be present in access_log_directory.	✓	true	Restart Replay if changed	Restart Replay if changed
Notifier section		✓			
kafka_format	Format of the Kafka message.	✓	1	✗	✓
kafka_endpoint	IP address and port of the Kafka endpoint.	✓	Retrieved from Manager API	✗	✓
kafka_topic	Name of the Kafka topic.	✓	Retrieved from Manager API	✗	✓

Element	Description	Optional	Default value	Restart scanner if changed	Restart notifier if changed
System section					
`kafka_username`	The username for authentication with Kafka. Note: For testing only. Not to be used in production.	✓	Supplied by user at prompt if you cannot retrieve from Manager API.	✗	✓
`kafka_password`	The password for authentication with Kafka. Note: For testing only. Not to be used in production.	✓	Supplied by user at prompt if cannot be retrieved from Manager API.	✗	✓
`kafka_pem`	The certificate PEM for authentication with Kafka. Must include '\n' characters to ensure correct formatting. Note: For testing only. Not to be used in production.	✓	Supplied by user at prompt if it cannot be retrieved from the system	✗	✓
Logging section					
`debug_log_max_bytes`	The scanner.debug and notifier.debug roll over when this size is reached.	✓	1,000,000	✓	✓
`debug_log_backup_count`	The number of scanner.debug and notifier.debug files to retain.	✓	10	✓	✓
`notification_log_max_b`	The notification.log rolls over when this size is reached.	✓	1,000,000	✓	✓
`notification_log_backup_count`	The number of notification.log files to retain.	✓	10	✓	✓
`notification_log_all`	Boolean value that controls the level of Notifier logging. When true: an entry is written to notification.log for message you send to the Kafka cluster. When false: only failed sends are written to notification.log.	✓	False	✗	✓
Root-level items					
`include_all_vaults`	Boolean value that determines whether all vaults in the dsNet are scanned. If false, the details of the vaults to be scanned must be specified in the 'vaults' element. Boolean value that determines whether custom metadata and content type are retrieved for each object by using individual HEAD requests.	✓	False	✓	✗
`has_custom_metadata`	This value is only relevant when a versioned vault is scanned. For IBM Cloud Object Storage systems, non-versioned vaults always require a HEAD request for every object. Can be overridden for each vault in the 'vaults' element.	✓	True	✓	✗
`override_warnings`	Boolean value that allows the scanner to run and ignore any warnings that are generated on start-up. For example, a warning is raised on start-up if versioning is suspended on a vault.	✓	False	✓	✗
`exclude_vaults`	Comma-separated list of vault names to be excluded from scanning, such as: `"exclude-vaults": ["COSVault", "COSVault-V"]`	✓	[] Empty list	✓	✗
`vaults`	List of vaults to be scanned. If `include_all_vaults` is true, the vaults list can be left empty. This list can be used to define more detailed scanning parameters for individual vaults. Any settings that are defined here take precedence over the settings that are described. Each element in the list contains: The `vault_name` is the name of the vault. The `has_custom_metadata` is an optional Boolean that overrides the `has_custom_metadata` that is described. The `prefix` is an optional string that is used to filter the objects or versions that are retrieved from the vault.	✓	Dependent on settings include_all_vaults and exclude_vaults	✓	✗

Replay performance

The number of requests that are issued by IBM Cloud® Object Storage Replay is throttled to ensure that overall dsNet performance remains at the agreed level.

You can control throttling by the number of settings in a configuration file. All settings are optional. The following screen shows an example of the default values.

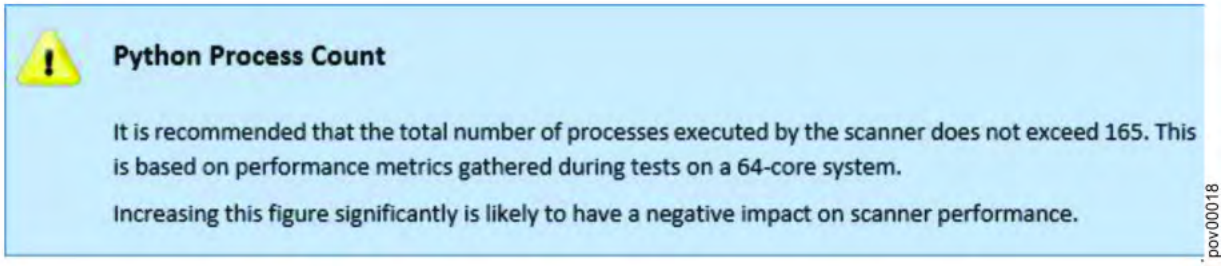
```
"replay": {
     "max_requests_per_second": 1000,
     "max_parallel_list": 10,
     "parallel_head_per_list": 15,
     "list_objects_size": 1000
}
```

Process count

The following list shows an example of how 161 processes are divided. Figure 1 shows a caution message of how the number of processes should not exceed 161.

- One main process
- 10 List worker processes
- 150 HEAD worker processes

Figure 1. Python process count



Maximum Replay performance

Replay and Notifier maximize performance on a 64 core Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz server is 2300 objects that are scanned and notified per second with a dsNet with 6 accessers and 12 slicestors under customer load at 50 percent capacity.

```
"replay": {
     "max_requests_per_second": 2300,
     "max_parallel_list": 10,
     "parallel_head_per_list": 15,
     "list_objects_size": 1000
 }
```

The recommendation is to start the replay at a rate of 1000 objects scanned per second. Measure the latency degradation of customer traffic and increase the scanning rate until the maximum acceptable degradation is reached.

One thousand objects per second on the net, which is a 5 - 27 percent increase of write operations, the latency (larger increase for smaller size files) and around 10 percent for read operations latency were measured.

At 2000 objects a second, a 10 - 50 percent increase of write operations latency and in the range 18 - 28 percent, and 10 percent for read operations latency were measured.

Replay tasks and vault settings

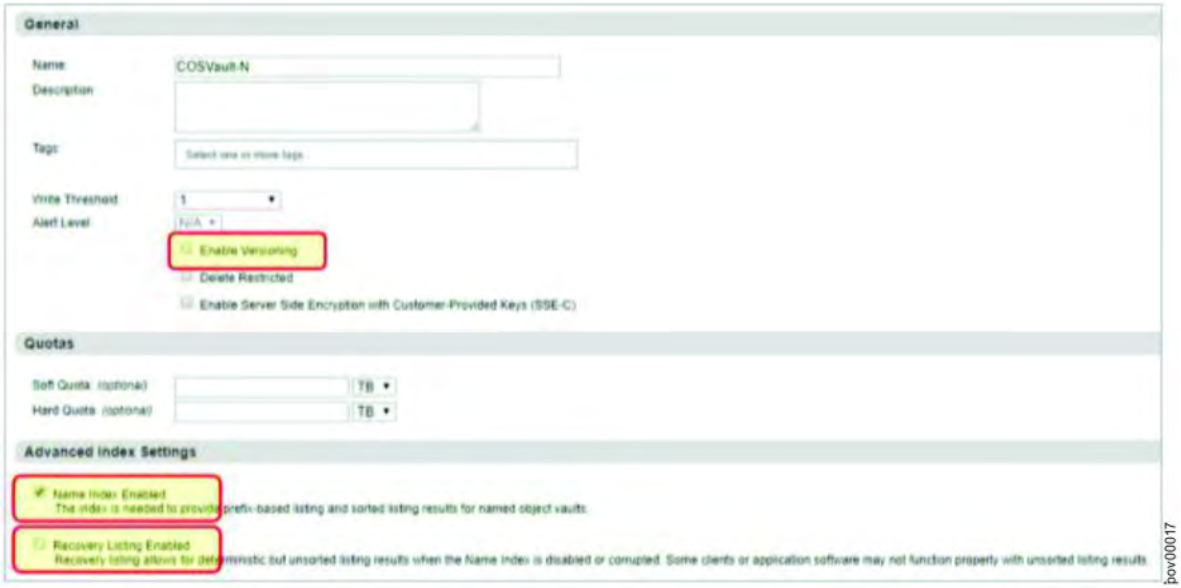
A few scenarios exist that prevent the Replay from operating correctly.

Certain combinations of the following IBM Cloud® Object Storage vault settings prevent the Replay from running a full scan:

- Vault versioning
- Name index
- Recover listing

Figure 1 shows settings for the first three items on the vault configuration page in the DsNet Manager user interface.

Figure 1. Settings for three items on the vault configuration page in the net Manager user interface



The scenarios that are invalid are reported at startup.
Remember: You must correct the scenarios before you can run the Replay.

If a scan does not complete successfully, make sure that you check the log file for errors and warnings. In some cases, you must modify the settings file as detailed in the errors and warning messages. The settings file is located at: /opt/ibm/metaocean/data/connections/cos/scan/scanner-settings.json

Table 1 shows the behavior for the Replay for different combinations of the four variables.

Table 1. Behaviors for Replay for four variables

ID	Name index	Recovery listing	Versioned	Cloud Object Storage Replay behavior
0	✗	✗	✗	☒ Stop start-up and report error in config file: Error: Objects cannot be listed because Name Index and Recovery Index are both disabled. You might enable Recovery Listing on the vault or add this vault to the `"exclude_vaults"` list in the configuration file. For example: `"exclude-vaults": ["vault-name"]"`
1	✗	✗	✓	☒ Stop start-up and report error in config file: Error: Objects cannot be listed because Name Index and Recovery Index are both unavailable. You might enable Recovery Listing on the vault or add this vault to the `"exclude_vaults"` list in the configuration file. For example: `"exclude-vaults": ["vault-name"]"`
2	✗	✓	✗	☑ Object Listing is run.
3	✗	✓	✓	☑ Object Listing runs. Only the most recent version of each object is listed. A warning is logged: Warning: Versions cannot be listed as Name Index is unavailable. An object scan is run and only the most recent version of each object is listed. You must add **`override_warnings: true`** in the configuration file to ignore this warning. Switching Name Index on does not enable scanning of a full version history. Objects created while Name Index is off is not present when it is enabled.
4	✓	✗	✗	☑ Object Listing is run.
5	✓	✗	✓	☑ Object Listing is run.
6	✓	✓	✗	☑ Object Listing is run.
7	✓	✓	✓	! Stop start-up and report warning: This is a versioned vault but version scanning is not possible as Recovery Listing is enabled. You might either disable Recovery Listing on the vault to allow version scanning, or rerun the Replay with the argument `override-warnings: true` to allow object scanning.

Important: You might receive system errors about records not being scanned to the database if you scan a IBM Cloud Object Storage vault with Name Index disabled and Recovery Listing enabled. You cannot specify a prefix listing on a vault that has Recovery Listing enabled (you cannot specify a blank prefix either).

Including and excluding vaults

You can set the vaults that you scan with various settings in the configuration file.

Use the following settings in the configuration file to scan the vaults:

- **`include_all_vaults`** (Boolean)
- **`exclude_vaults`** (List)
- **`vaults`** (Dictionary)

When **`include_all_vaults`** is true, all vaults in the system are scanned except for any vaults specified in the **`exclude_vaults`** list.

You might consider **`exclude_vaults`** a list of vaults to ignore and **`vaults`** is a list that specifies details of individual vaults to be scanned.

If **`include_all_vaults`** is true and the vaults list is populated, the list of vaults that are scanned is the superset of all vaults that are returned by the Manager that are merged with the vaults list from the config file.

An error is raised and the Scanner aborts on start-up if the same vault appears in both **`vaults`** and **`exclude_vaults`**.

Mirror, Proxy, Data Migration

IBM Cloud® Object Storage Scanner does not support scanning of the following:

- Mirrored vaults
- Proxy vaults
- Vaults that are set up for migration

Any vaults of these types are ignored by the scanner and a warning logged in the debug log.

Examples for including and excluding vaults

To summarize the rules for including and excluding vaults, following are some examples:

Example 1

- The system contains 1000 vaults.
- Five of the 1000 vaults are management vaults (named mgmt-1 to mgmt-5).

- The scan includes all vaults except the management vaults.

```
"include_all_vaults": true,
"exclude-vaults": ["mgmt-1", "mgmt-2", "mgmt-3", "mgmt-4", "mgmt-5"]
```

Example 2

- The system contains 1000 vaults.
- 5 of the 1000 vaults are management vaults (named mgmt-1 to mgmt-5).
- The scan includes all vaults except the management vaults.
- The scan includes a filter for scanning a vault that is named vault-x.
- The scan includes only a scan of the objects whose key starts with **production/finance**.

```
"include_all_vaults": true,
"exclude-vaults": ["mgmt-1", "mgmt-2", "mgmt-3", "mgmt-4", "mgmt-5"],
"vaults": [
     {"vault_name":"vault-x", "prefix":"production/finance"}
  ]
```

Example 3

- The system contains 1000 vaults.
- 5 of the 1000 vaults are management vaults (named mgmt-1 to mgmt-5).
- The scan includes all vaults except the management vaults.
- The scan includes a filter for scanning a vault that is named vault-x.
- The scan includes only a scan of the objects whose key starts with **production/finance** or **production/marketing**.

```
"include_all_vaults": true,
"exclude-vaults": ["mgmt-1", "mgmt-2", "mgmt-3", "mgmt-4", "mgmt-5"],
"vaults": [
     {"vault_name":"vault-x", "prefix":"production/finance"},
     {"vault_name":"vault-x", "prefix":"production/marketing"}
   ]
```

Example 4

- The system contains 1000 vaults.
- Run a test on three vaults named vault-a, vault-b, and versioned-vault-c.
- Run a scan on versioned-vault-c and issue LIST requests. Do not issue HEAD requests because the objects do not have custom amz headers.

```
"include_all_vaults": false,
"vaults": [
     {"vault_name":"vault-a"
     {"vault_name":"vault-b"
     {"vault_name":"vault-c",  "has_custom_metadata":false}
   ]
```

Stats files

The IBM Cloud® Object Storage Scanner tracks each LIST process status to a stats file.

During a scan, the Scanner runs multiple processes. Each `LIST` processes and tracks the progress, saves the **`next_key`**, and optionally the **`next_version`** to a stats file named task.stats that is stored with the log files in the /opt/ibm/metaocean/data/connections/cos/replay/output/data directory.

```
{
     "estimated_object_count": 1000,
     "list_objects_size": 100,
      "next_key": "",
      "next_version": "",
      "prefix": "",
      "scan_type": "Object Scan",
      "status": "Complete",
      "total_bytes_output": 1126809,
      "total_bytes_scanned": 1126809,
      "total_objects_output": 47,
      "total_objects_scanned": 47,
      "vault_name": "dsmgmt-sp1",
      "vault_uuid": "868daa21-9e56-4c41-b6fd-845a4c85cea9"
}
```

From the Scanner, you can start, stop, recover files from a crash, and restart at the point where the scan was interrupted.

When you start the scanner:

1. Processing of the Scanner continues from **`next_key`** and **`next_version`**.
2. Queue of the Notifier is optimized by reloading from the files in the data folder instead of requerying the dsNet.
3. Batches that were processed partially are reprocessed. Duplicate Kafka notifications might occur, but are handled safely by the IBM Data Cataloging system.

Replay

When a severe outage occurs and causes the loss of notifications sent by the system to IBM Data Cataloging, the IBM Cloud® Object Storage Scanner Replay feature can be used to recover lost notifications.

Replay parses the access logs of a system and reconstitutes the notifications. Also, the Notifier can resend the notifications.

Initialization for Replay

During the startup, Replay reads the configuration file and issues requests to the Manager of the dsNet device similar to the Scanner.

Data from the configuration file is validated to ensure that appropriate permissions are granted in the dsNet. This allows access to management vaults and regular vaults. Startup errors or warnings are logged and printed to the console.

After initialization, Replay extracts accesser log files from the management vaults of dsNet and enables Replay to process and write notifications to the output directory.

Error conditions

Sometimes Replay does not have enough information to replay the original notification. If this occurs, you must fix the problems manually.

For example, if vault versioning was suspended when you made the request and you receive an s3 DeleteObject for an object or delete marker, the following error is logged:

```
error_code=True, error_description="Delete operation with [no version_id|null
version_id|version_id] for vault with versioning = [suspended/enabled]"
```

The error message displays because Replay cannot distinguish when a notification with s3:CreateDeleteMarker or s3:CreateDeleteMarker:NullVersionDeleted is sent.

If vault versioning is disabled, and an s3 PutObject request is received for an object that is deleted, the following error is logged:

```
error_code=404, error_message="Not Found"
```

The error message displays because Replay cannot determine the tag of the object that was deleted.

Output

Messages are batched by 1,000 or to the Scanner list objects size configuration setting, if specified.

The messages are written to the output folder with the same Notification format used by the Scanner.

```
{
     "system_name": "Test",
     "object_etag": "\"de37d2cee49596916f62a233dfc790a4\"",
     "request_time": "2018-09-24T18:49:29.383Z",
     "format": 1,
     "bucket_uuid": "ac89915b-d4ec-7ff1-00be-9c32b2aca580",
     "system_uuid": "f7d033c2-9066-499a-a883-829860d4d865",
     "object_length": "12319",
     "object_name": "test_version",
     "bucket_name": "vault3",
     "content_type": "binary/octet-stream",
     "request_id": "17451c3d-e81e-40ed-939a-4534780daaa8",
     "operation": "s3:PutObject"
}
```

If an error occurs, the error messages are written to the /opt/ibm/metaocean/data/connections/cos/replay/output/data/access_log_error/ directory. Take note of the extra **error_code** and **error_description** elements.

```
{
     "system_name": "Test",
     "object_version": "null",
     "request_time": "2018-09-24T17:07:59.471Z",
     "format": 1,
     "bucket_uuid": "ac89915b-d4ec-7ff1-00be-9c32b2aca580",
     "system_uuid": "f7d033c2-9066-499a-a883-829860d4d865",
     "object_length": "12319",
     "object_name": "object5.2",
     "bucket_name": "vault3",
     "request_id": "ebc472a1-f955-4605-895b-840867b12e01",
     "operation": "s3:PutObject",
     "error_description": "Not Found",
     "error_code": 404
}
```

Renaming a vault for Replay

When you rename a vault, it is possible that Replay can abort.

Replay aborts when you:

- Delete the vault.
- Rename the vault.
- Discover that the read permission is revoked for the credentials that are supplied by the operator or manager API.

All other scans of a vault continue scanning until complete.

You can find the details of the errors that include stack trace in the replay.debug file in the /opt/ibm/metaocean/data/connections/cos/replay/debug/replay/[timestamp] directory.

Starting the Replay

The guidelines and rules for using Replay are documented in this topic.

To start Replay, run the following command:

```
cos-replay
```

The following rules apply for Replay:

- Configure Replay according to the guidelines in Table 1.
- Replay component requires `min_utc` and `max_utc` time stamps defined in the Configuration file.
- Only notifications sent between `min_utc` and `max_utc` are parsed and replayed.
- Replay automatically shuts down when all accesser logs are downloaded and processed. The message Complete Replay Process appears in the console.

This is an example of how to start Replay:

```
Starting COS Replay - Version 0.1 Log file and config file are in directory /Users/weebrew/
Documents/Development/ibmworkspace/cosscanner/output/debug/scanner/20180925-125528-232131
Starting Accessor Log Extraction Downloading files...
('Downloaded', 10, 'of', 36)
('Downloaded', 20, 'of', 36)
('Downloaded', 30, 'of', 36)
Download complete.
Total files: 36
Complete Accessor Log Extraction
Starting Replay process
Complete Replay process
```

Debug mode for Replay

Run Replay in debug mode to troubleshoot problems.

To start debug mode, run the following command:

```
cos-replay --log=DEBUG
```

Running debug mode creates large log files and creates a significant drop in performance. Do not run debug mode for long periods especially when you are in production mode.

Notifier

The Notifier is the component that reads the JSON notifications that are written by Scanner or Replay and sends notifications to the Kafka cluster.

When notifications are acknowledged by Kafka, the Notifier moves the file to the archive folder.

On start-up, Notifier calls the Manager API and retrieves details of any Notification Service Configurations (NSC) configured in the dsNet for IBM Data Cataloging. If more than one is found, the first one is used.

Retrieval of NSCs is overridden by defining the details of the Kafka configuration in the config file.

```
 "notifier":{
   "kafka_format": 1,
   "kafka_endpoint": "192.168.1.34:9092",
   "kafka_topic": "cos-le-connector-topic"
}
```

Limitations

Limitations apply when the Notifier uses a Kafka configuration retrieved from the Manager API.

- If more than one NSC exists, the first one is used for all vaults.
- If more than one host name is defined in the NSC, the first one is used for all vaults.

Starting the Notifier

Running the Notifier has rules and limitations.

To start the Notifier, run the following command:

```
cos-notify
```

After you start the Notifier, you are prompted for security credentials for the manager API and Kafka cluster.

```
Starting COS Notifier - Version 0.1
Enter the Manager API username: admin
Enter the Manager API password:
Enter the Kafka username: cos
Enter the Kafka password:
Enter the Kafka pem:
Creating Kafka producer...
Done
Notifier is running
Log file and config file are in directory C:\dev\cos-scanner\output\debug\notifier\20180912-121641-283000
Checking for files in   \data
- 11 files found Checking for files in output\data
- 256 files found
```

Rules and limitations

The following rules and limitations apply to the Notifier:

- You cannot start the Notifier in the background because the Notifier requires user input at the terminal window.
- You can stop the Notifier and force the Notifier to run in the background.
- The passwords and pem do not display when you type and paste the passwords in the console.
- The certificate pem is approximately 1600 characters. If you use an SSH connection, the certificate pem might be truncated to 1000 characters.
- If the number is truncated to 1000 characters, include the certificate pem in the config file.

Notifier operation

The Notifier enumerates and processes all .log files in the Scanner data directory.

After all files are processed, the Notifier repeats the process so that new .log files that are generated by the Scanner are processed. The Notifier sleeps repeatedly in 1-second intervals if no new files are found in the /opt/ibm/metaocean/data/connections/cos/replay/output/data directory.

The Notifier does not automatically shut down. The Notifier continues to monitor the Scanner data directory for new .log files. Monitor the progress of the Notifier by using the status report. When the operator or administrator determines that all scanned objects are submitted successfully to the IBM Data Cataloging, shut down the Notifier by using the kill switch.

Stopping the Notifier

You might need to stop and restart the Notifier.

Before you stop and restart the Notifier:

1. Create a file named kill.notifier in the /opt/ibm/metaocean/data/connections/cos/replay/output/command directory.
2. Ensure that the processing of any batches is complete before you stop the Notifier.

Stopping the Notifier displays the following output:

The shutdown is complete when the "**Shutdown is complete**" message displays.

```
Starting COS Notifier - Version 0.1
Enter the Manager API username: admin
Enter the Manager API password:
Enter the Kafka username: cos Enter the Kafka password:
Enter the Kafka pem: Creating Kafka producer...
Done
Notifier is running
Log file and config file are in directory C:\dev\cos-scanner\output\
   debug\notifier\20180912-121641-283000
Checking for files in output\data
- 11 files found Checking for files in output\data
- 256 files found Detected the kill trigger file. Shutting down...
Shutdown is complete
```

Restarting the Notifier

When you stop the Notifier following a shutdown with the kill.notifier file, you must rename the file manually or delete the file before you do a restart.

If you do not rename or delete the kill.notifier file, the system finds the file and displays the following message:

```
C:\dev\cos-scanner>python main_notifier.py Starting COS Notifier - Version 0.1
The file 'kill.notifier' is preventing the notifier from running.
```

```
You should delete or rename the file and re-start the notifier.
File location: 'C:\dev\cos-scanner\command\kill.notifier
```

The Scanner and Notifier are separate solutions that share the config file. The files operate independently, so you can start and stop either file independently any time.

Progress report

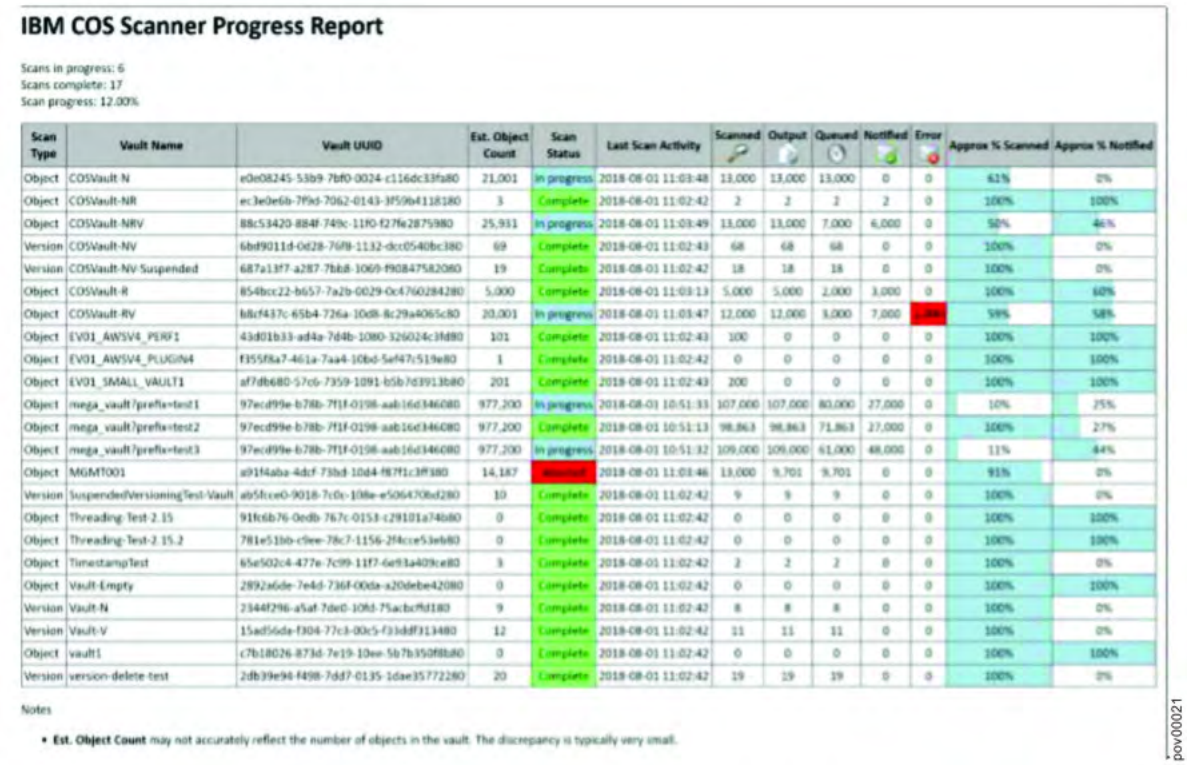
The Progress Report provides an instant snapshot of status for the Scanner and Notifier.

To create a progress report, run the following command:

```
cos-report
```

The progress report displays in plain text format to the console in a static HTML file that is named: /opt/ibm/metaocean/data/connections/cos/replay/output/cos-scanner-report.html
If a progress report exists, the new progress report overwrites the existing progress report. See Figure 1.

Figure 1. IBM Cloud Object Storage Scanner progress report



IBM COS Scanner Progress Report

Scans in progress: 6
Scans complete: 17
Scan progress: 12.00%

Scan Type	Vault Name	Vault UUID	Est. Object Count	Scan Status	Last Scan Activity	Scanned	Output	Queued	Notified	Error	Approx % Scanned	Approx % Notified
Object	COSVault-N	e0e08245-53b9-7bf0-0024-c116dc33fa80	21,001	In progress	2018-08-01 11:03:48	13,000	13,000	13,000	0	0	61%	0%
Object	COSVault-NR	ec3e0e6b-7f9d-7062-0143-3f59b4118180	3	Complete	2018-08-01 11:02:42	2	2	2	2	0	100%	100%
Object	COSVault-NRV	88c53420-884f-749c-11f0-f27fe2875980	25,931	In progress	2018-08-01 11:03:49	13,000	13,000	7,000	6,000	0	50%	46%
Version	COSVault-NV	6bd9011d-0d28-76f8-1132-dcc0540bc380	69	Complete	2018-08-01 11:02:43	68	68	68	0	0	100%	0%
Version	COSVault-NV-Suspended	687a13f7-a287-7bb8-1069-f90847582080	19	Complete	2018-08-01 11:02:42	18	18	18	0	0	100%	0%
Object	COSVault-R	854bcc22-b657-7a2b-0029-0c4760284280	5,000	Complete	2018-08-01 11:03:13	5,000	5,000	2,000	3,000	0	100%	60%
Object	COSVault-RV	b8cf437c-65b4-726a-10d8-8c29a4065c80	20,001	In progress	2018-08-01 11:03:47	12,000	12,000	5,000	7,000	[illegible]	59%	58%
Object	EV01_AWSV4_PERF1	43d01b33-ad4a-7d4b-1080-326024c3fd80	101	Complete	2018-08-01 11:02:43	100	0	0	0	0	100%	100%
Object	EV01_AWSV4_PLUGIN4	f355f8a7-461a-7aa4-10bd-5ef47c519e80	1	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Object	EV01_SMALL_VAULT1	af7db680-57c6-7359-1091-b5b7d3913b80	201	Complete	2018-08-01 11:02:43	200	0	0	0	0	100%	100%
Object	mega_vault?prefix=test1	97ecd99e-b78b-7f1f-0198-aab16d346080	977,200	In progress	2018-08-01 10:51:33	107,000	107,000	80,000	27,000	0	10%	25%
Object	mega_vault?prefix=test2	97ecd99e-b78b-7f1f-0198-aab16d346080	977,200	Complete	2018-08-01 10:51:13	98,863	98,863	71,863	27,000	0	100%	27%
Object	mega_vault?prefix=test3	97ecd99e-b78b-7f1f-0198-aab16d346080	977,200	In progress	2018-08-01 10:51:32	109,000	109,000	61,000	48,000	0	11%	84%
Object	MGMT001	a9114aba-4dcf-73bd-10d4-f87f1c3ff380	14,187	Aborted!	2018-08-01 11:03:46	13,000	9,701	9,701	0	0	91%	0%
Version	SuspendedVersioningTest-Vault	ab5fcce0-9018-7c0c-108e-e5064708d280	10	Complete	2018-08-01 11:02:42	9	9	9	0	0	100%	0%
Object	Threading-Test-2.15	91fc6b76-0edb-767c-0153-c29101a740d0	0	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Object	Threading-Test-2.15.2	781e51bb-c9ee-78c7-1156-2f4cce53eb80	0	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Object	TimestampTest	65e502c4-477e-7c99-11f7-6e93a409ce80	3	Complete	2018-08-01 11:02:42	2	2	2	0	0	100%	0%
Object	Vault-Empty	2892a6de-7e4d-736f-00da-a20debe42080	0	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Version	Vault-N	2344f296-a5af-7de0-10fd-75acbcffd180	9	Complete	2018-08-01 11:02:42	8	8	8	0	0	100%	0%
Version	Vault-V	15ad56da-f304-77c3-00c5-f33ddf313480	12	Complete	2018-08-01 11:02:42	11	11	11	0	0	100%	0%
Object	vault1	c7b18026-873d-7e19-10ee-5b7b350f8b80	0	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Version	version-delete-test	2db39e94-f498-7dd7-0135-1dae35772280	20	Complete	2018-08-01 11:02:42	19	19	19	0	0	100%	0%

Notes

- **Est. Object Count** may not accurately reflect the number of objects in the vault. The discrepancy is typically very small.

pov00021

See Table 1 for a description of information in the progress report.

Table 1. Description for IBM Cloud Object Storage Scanner progress report

Column name	Description
Scan Type	Either Object or Version. Non-Versioned vaults show Object. Versioned vaults show Version. However, there are some exceptions. If the Name Index for a versioned vault is unavailable but Recovery Listing is enabled, an object scan might be run. The user is alerted that an object scan can be done, but this object scan requires changes to the configuration file.
Vault Name	The name of the vault. Any prefix that is defined in the configuration file is also shown. Example: `mega_vault?prefix=test`
Vault UUID	The UUID of the vault.
Estimated Object Count	The estimated number of objects in the vault, as reported by the Manager API. This value is refreshed from the Manager API each time the scanner is started, regardless of the status of each scan. Given that the number of objects in each vault might be constantly changing, the number of objects that are reported in this column becomes out of date during long running scans. Note: This issue affects only the status report but does not affect the data integrity of the Scanner.

Column name	Description
Scan Status	Shows the status of the scanner. Not started The task is queued but not started. In progress The task is running. Complete The task finished. Aborted The task encountered an unrecoverable error and aborted. Shut down the Scanner and the debug file, and inspect the file to investigate the problems. After you resolve the problems, restart the Scanner. The debug file is in the /opt/ibm/metaocean/data/connections/cos/replay/output/data/<vault-name>/<prefix> directory. For each vault, see the /opt/ibm/metaocean/data/connections/cos/replay/output/data/<vault-name>/<prefix> directories.
Last scan activity	The last time data was retrieved from the vault.
Scanned	Number of objects or versions that are scanned. For a versioned vault, this value shows a figure that is higher than the Estimated Object Count.
Output	Number of objects or versions that scanned AND whose `LastModified` time stamp is inside the time window that is defined in the configuration file. The figure in the column is Queued + Notified + Error.
Queued	Number of objects or versions that are Output and are waiting to be sent to the Kafka cluster.
Notified	Number of objects or versions that are submitted successfully to the Kafka cluster.
Error	Number of objects or versions that failed to send to the Kafka cluster. Details of all errors are logged to notifier.debug.
Approximate percentage scanned	Scanned as a percentage of Est. Object Count. The cell background shows a progress bar.
Approximate percentage scanned	Notified as a percentage of Output. The cell background shows a progress bar.

Table 2. What is reported beneath the report title

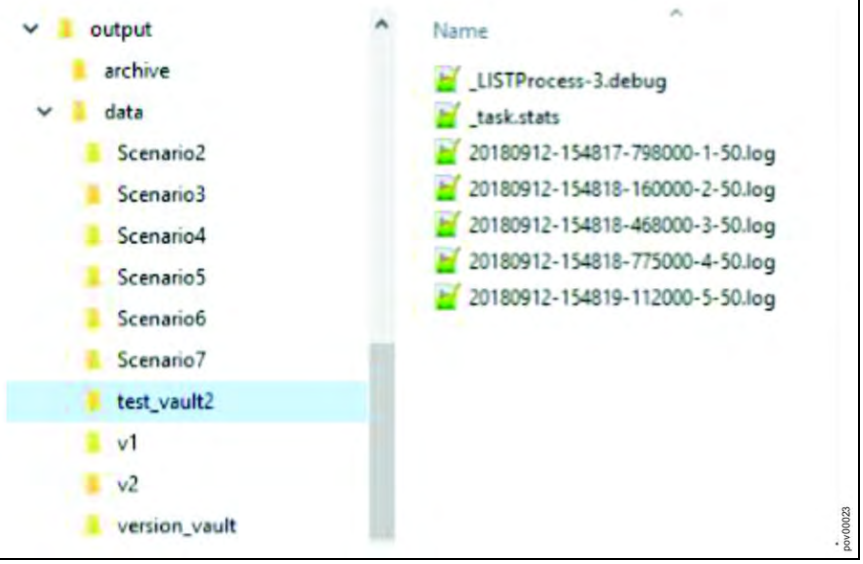
Measure	Description
Scans in progress	Number of scans with the status `"In Progress"`. Applies to Scanner only.
Scans complete	Number of scans with the status `"Complete"`. Applies to Scanner only.
Scan progress	Sum (number of objects scanned) as a percentage of sum (estimated object count).

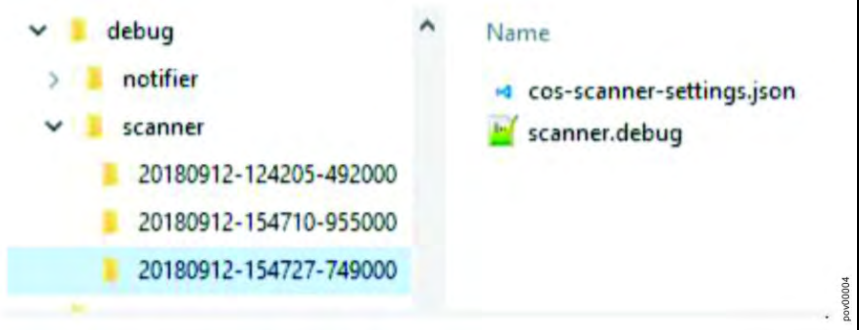
Logging

You can view the list of directories that are generated by scanner, notifier, and replay.

Table 1 lists the directories that are generated on start-up by the Scanner, Notifier, and Replay.

Table 1. List of directories generated by scanner, notifier, and replay

Directory	Description
For IBM Data Cataloging: `/opt/ibm/metaocean/data/connections/cos/replay/output/data`	Contains .log files (Kafka messages), stats files and debug information for each scanned vault. 

Directory	Description
For IBM Data Cataloging: `/opt/ibm/metaocean/data/connections/cos/replay/output/debug/[scanner|replay]/`	Contains Scanner and Replay debug or troubleshooting information. A new subdirectory is created each time the Scanner and Replay starts. Each subdirectory contains a copy of the configuration file and scanner.debug (replay.debug). 
For IBM Data Cataloging: `/opt/ibm/metaocean/data/connections/cos/replay/output/debug/notifier`	Contains Notifier debug or troubleshooting information. A new subdirectory is created each time the Notifier starts. It contains a copy of the configuration file and notifier.debug. Same directory-naming convention as shown for the scanner. The notifier.debug rolls over when it reaches a predefined size as defined in the configuration file. See Configuration file.
For IBM Data Cataloging: `/opt/ibm/metaocean/data/connections/cos/replay/output/archive/`	Contains all .log files that are successfully processed by the Notifier, and their contents are successfully submitted to the Kafka cluster. Remember: Files in this directory are truncated – they contain only the object key and they can also contain the version.
For IBM Data Cataloging: `/opt/ibm/metaocean/data/connections/cos/replay/output/error/`	Contains any .log files that failed to submit to the Kafka cluster.
For IBM Data Cataloging: `/opt/ibm/metaocean/data/connections/cos/replay/output/notification_log/`	The notification.log file contains details of any errors (including stack trace) that occur when it attempts to send notifications to the Kafka cluster. If `logging/notification_log_all` is true in the config file, all successful sends are also logged. The notification.log file rolls over when it reaches a predefined size as defined in the configuration file. See Configuration file.

IBM Cloud Object Storage Scanner output data

The Scanner generates a directory beneath the output data directory for each vault or vault prefix as defined in the configuration file.

The /opt/ibm/metaocean/data/connections/cos/replay/output/data directory is the Scanner output data directory.

The following screen shows an example of a configuration file and also shows that all vaults are scanned, but mega_vault has four separate prefixes that are defined winch means the four scans of the vault occurred.

```
"include_all_vaults": true,
  "vaults": [
  {"vault_name": "mega_vault", "prefix": "main/production/finance"},
  {"vault_name": "mega_vault", "prefix": "main/production/sales"},
  {"vault_name": "mega_vault", "prefix": "main/production/marketing"},
  {"vault_name": "mega_vault", "prefix": "main/production/hr"}
]
```

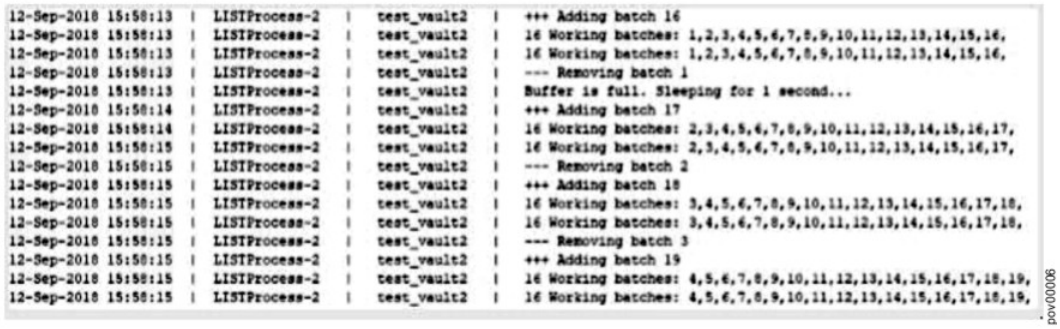
Figure 1 shows the directory structure.

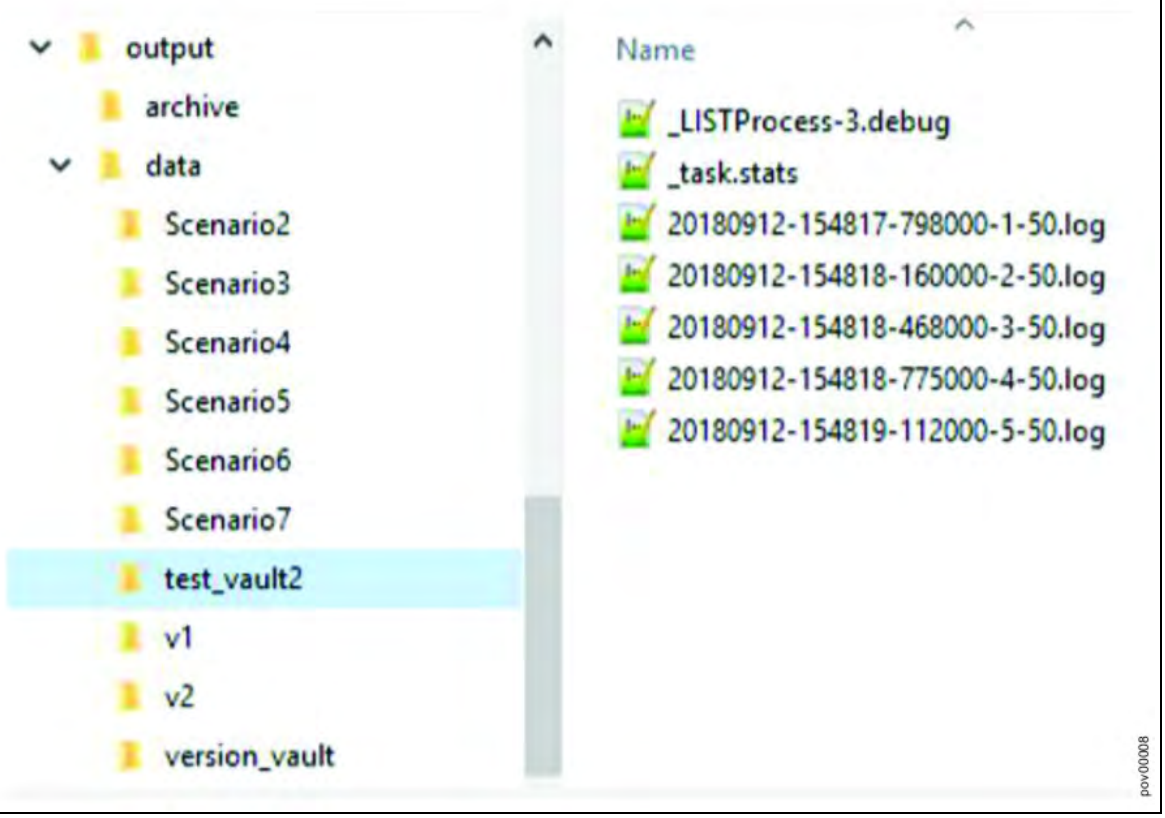
Figure 1. Directory structure from the configuration file



The status and progress of each scan must be maintained so a separate directory structure is created for each scan. Table 1 shows the leaf directories that contain the file names and description.

Table 1. Leaf directory file names

File name	Description
_LISTProcessN.debug	The N in the file name is different for each process (0 - 9 if there are 10 processes). Contains detailed debug information and details of any errors that are encountered when you scan the vault. Figure 2 shows an example of running in debug mode. Figure 2. Example of running in debug mode 

File name	Description
task.stats	Scanner starts in JSON format for a single vault. Updated following successful processing of each batch of objects. "estimated_object_count": 1718, "list_objects_size": 50, "next_key": "", "next_version": "", "prefix": "", "scan_type": "Object Scan", "status": "Complete", "total_bytes_output": 45257, "total_bytes_scanned": 45257, "total_objects_output": 1717, "total_objects_scanned": 1717, "vault_name": "test_vault2", "vault_uuid": "06c1641d-082f-7ba2-011b-c7550651a780" 
*.log	The Scanner creates multiple .log files for each vault. Each .log file contains up to 1000 Kafka messages, ready to be submitted to the Kafka cluster by the Notifier. The naming convention for the log files is **<date>-<time>-<milliseconds>-<batch number>-<number of messages in file>.log** 

Appendix

The appendix shows an example of a log file and examples of Scanner debug data.

Log file

Figure 1 shows an example with extra line breaks.

Figure 1. Example of a log file

```
{"system_name": "Test", "object_version": "38d811e6-dba1-4830-859d-6275f2016bc3", "object_etag":
"\"73b8c5dcca9cc6aab928396a2d98a340\"", "request_time": "2018-08-24T18:39:16Z", "format": 1, "bucket_uuid": "cbc03649-
d218-727d-10ec-df8c22873280", "system_uuid": "f7d033c2-9066-499a-a883-829860d4d865", "meta_headers": [], "object_length":
12, "object_name": "my-object-black", "bucket_name": "version_vault", "content_type": "text/plain", "request_id":
"7ed39768-1184-4d5f-8c0c-7912515bf8fe", "operation": "s3:PutObject"}

{"system_name": "Test", "object_version": "8a68208b-9b5f-4107-9eae-fa08200c7913", "object_etag":
"\"73b8c5dcca9cc6aab928396a2d98a340\"", "request_time": "2018-08-24T18:39:15Z", "format": 1, "bucket_uuid": "cbc03649-
d218-727d-10ec-df8c22873280", "system_uuid": "f7d033c2-9066-499a-a883-829860d4d865", "meta_headers": [], "object_length":
12, "object_name": "my-object-black", "bucket_name": "version_vault", "content_type": "text/plain", "request_id":
"6ed2a774-f384-4cba-96fd-81dfbb681482", "operation": "s3:PutObject"}

{"system_name": "Test", "object_version": "322d9ed2-ca86-4efd-b95a-a2467ded9202", "object_etag":
"\"73b8c5dcca9cc6aab928396a2d98a340\"", "request_time": "2018-08-24T18:39:15Z", "format": 1, "bucket_uuid": "cbc03649-
d218-727d-10ec-df8c22873280", "system_uuid": "f7d033c2-9066-499a-a883-829860d4d865", "meta_headers": [], "object_length":
12, "object_name": "my-object-black", "bucket_name": "version_vault", "content_type": "text/plain", "request_id":
"ced1de9e-e62f-4966-b803-7aff3ddab245", "operation": "s3:PutObject"}

{"system_name": "Test", "object_version": "0e2b0369-a506-4ce3-8336-ea42eae16489", "object_etag":
"\"73b8c5dcca9cc6aab928396a2d98a340\"", "request_time": "2018-08-24T18:39:15Z", "format": 1, "bucket_uuid": "cbc03649-
d218-727d-10ec-df8c22873280", "system_uuid": "f7d033c2-9066-499a-a883-829860d4d865", "meta_headers": [], "object_length":
12, "object_name": "my-object-black", "bucket_name": "version_vault", "content_type": "text/plain", "request_id":
"ee5b8ded-86af-4e9e-9054-8902f7a7a5c0", "operation": "s3:PutObject"}

{"system_name": "Test", "object_version": "41518cb8-632f-45c4-989b-44b4d2b19b2e", "object_etag":
"\"73b8c5dcca9cc6aab928396a2d98a340\"", "request_time": "2018-08-24T18:39:15Z", "format": 1, "bucket_uuid": "cbc03649-
d218-727d-10ec-df8c22873280", "system_uuid": "f7d033c2-9066-499a-a883-829860d4d865", "meta_headers": [], "object_length":
12, "object_name": "my-object-black", "bucket_name": "version_vault", "content_type": "text/plain", "request_id":
"5adaa4d9-7aa6-4c46-bcd1-1260c074a972", "operation": "s3:PutObject"}

{"system_name": "Test", "object_version": "c75c6faf-e7a9-41ce-a576-6bfc9d374dfc", "object_etag":
"\"73b8c5dcca9cc6aab928396a2d98a340\"", "request_time": "2018-08-24T18:39:14Z", "format": 1, "bucket_uuid": "cbc03649-
d218-727d-10ec-df8c22873280", "system_uuid": "f7d033c2-9066-499a-a883-829860d4d865", "meta_headers": [], "object_length":
12, "object_name": "my-object-black", "bucket_name": "version_vault", "content_type": "text/plain", "request_id":
"35aa2dee-8700-4cfb-a62e-dc7f7ec7b8e1", "operation": "s3:PutObject"}
```


pov00009

Scanner debug data

Figure 2, Figure 3, Figure 4, and Figure 5 show that throttling settings are logged, and multiple HEAD processes are started for each LIST process.

Figure 2. Scanner debug

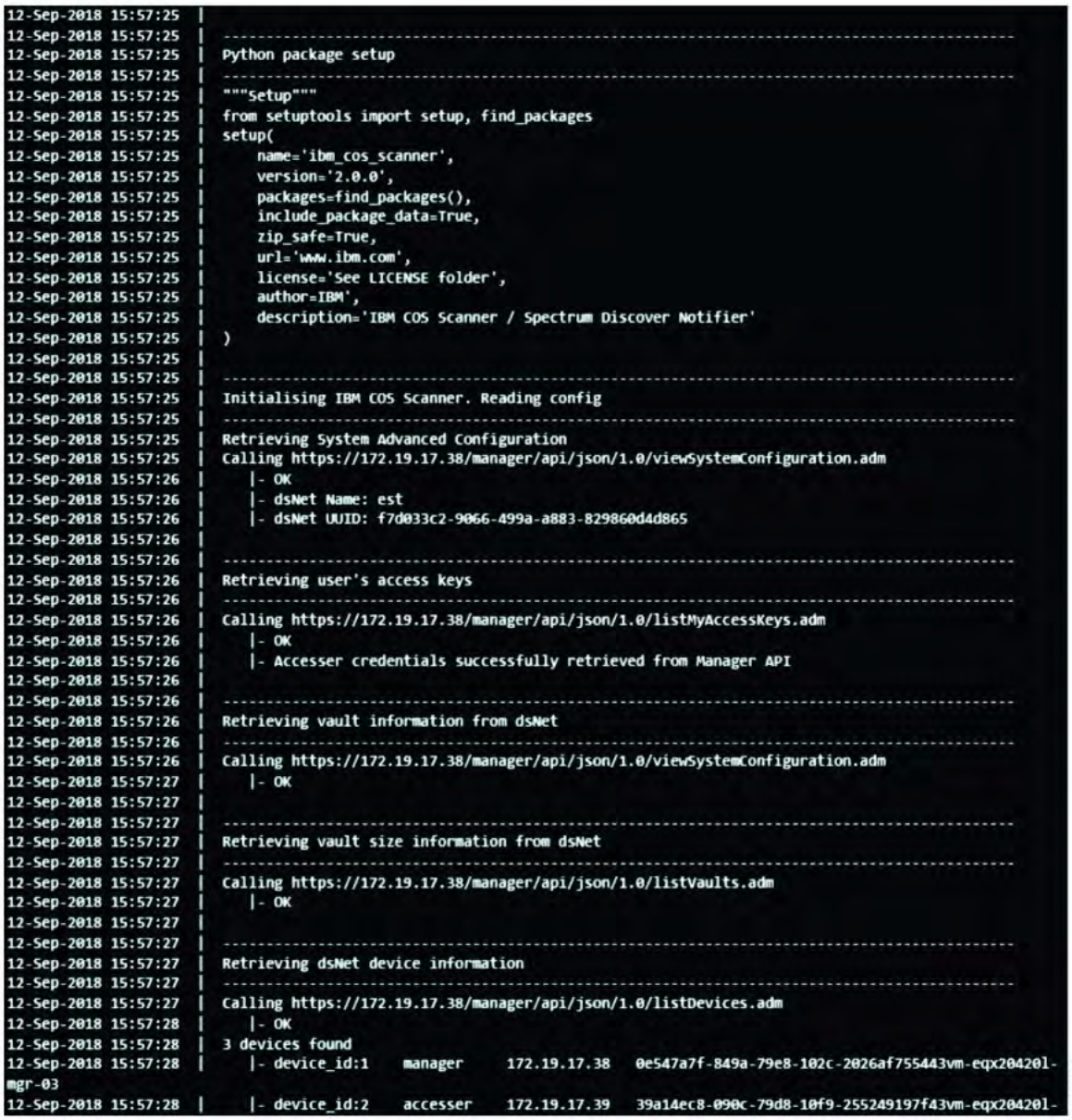


Figure 3. Scanner debug (continued)

```
12-Sep-2018 15:57:25 |
12-Sep-2018 15:57:25 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:25 |   Python package setup
12-Sep-2018 15:57:25 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:25 |   """Setup"""
12-Sep-2018 15:57:25 |   from setuptools import setup, find_packages
12-Sep-2018 15:57:25 |   setup(
12-Sep-2018 15:57:25 |       name='ibm_cos_scanner',
12-Sep-2018 15:57:25 |       version='2.0.0',
12-Sep-2018 15:57:25 |       packages=find_packages(),
12-Sep-2018 15:57:25 |       include_package_data=True,
12-Sep-2018 15:57:25 |       zip_safe=True,
12-Sep-2018 15:57:25 |       url='www.ibm.com',
12-Sep-2018 15:57:25 |       license='See LICENSE folder',
12-Sep-2018 15:57:25 |       author=IBM',
12-Sep-2018 15:57:25 |       description='IBM COS Scanner / Spectrum Discover Notifier'
12-Sep-2018 15:57:25 |   )
12-Sep-2018 15:57:25 |
12-Sep-2018 15:57:25 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:25 |   Initialising IBM COS Scanner. Reading config
12-Sep-2018 15:57:25 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:25 |   Retrieving System Advanced Configuration
12-Sep-2018 15:57:25 |   Calling https://172.19.17.38/manager/api/json/1.0/viewSystemConfiguration.adm
12-Sep-2018 15:57:26 |      |- OK
12-Sep-2018 15:57:26 |      |- dsNet Name: est
12-Sep-2018 15:57:26 |      |- dsNet UUID: f7d033c2-9066-499a-a883-829860d4d865
12-Sep-2018 15:57:26 |
12-Sep-2018 15:57:26 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:26 |   Retrieving user's access keys
12-Sep-2018 15:57:26 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:26 |   Calling https://172.19.17.38/manager/api/json/1.0/listMyAccessKeys.adm
12-Sep-2018 15:57:26 |      |- OK
12-Sep-2018 15:57:26 |      |- Accesser credentials successfully retrieved from Manager API
12-Sep-2018 15:57:26 |
12-Sep-2018 15:57:26 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:26 |   Retrieving vault information from dsNet
12-Sep-2018 15:57:26 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:26 |   Calling https://172.19.17.38/manager/api/json/1.0/viewSystemConfiguration.adm
12-Sep-2018 15:57:27 |      |- OK
12-Sep-2018 15:57:27 |
12-Sep-2018 15:57:27 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:27 |   Retrieving vault size information from dsNet
12-Sep-2018 15:57:27 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:27 |   Calling https://172.19.17.38/manager/api/json/1.0/listVaults.adm
12-Sep-2018 15:57:27 |      |- OK
12-Sep-2018 15:57:27 |
12-Sep-2018 15:57:27 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:27 |   Retrieving dsNet device information
12-Sep-2018 15:57:27 |   ----------------------------------------------------------------------------------------
12-Sep-2018 15:57:27 |   Calling https://172.19.17.38/manager/api/json/1.0/listDevices.adm
12-Sep-2018 15:57:28 |      |- OK
12-Sep-2018 15:57:28 |   3 devices found
12-Sep-2018 15:57:28 |      |- device_id:1    manager     172.19.17.38   0e547a7f-849a-79e8-102c-2026af755443vm-eqx20420l-
mgr-03
12-Sep-2018 15:57:28 |      |- device_id:2    accesser    172.19.17.39   39a14ec8-090c-79d8-10f9-255249197f43vm-eqx20420l-
```

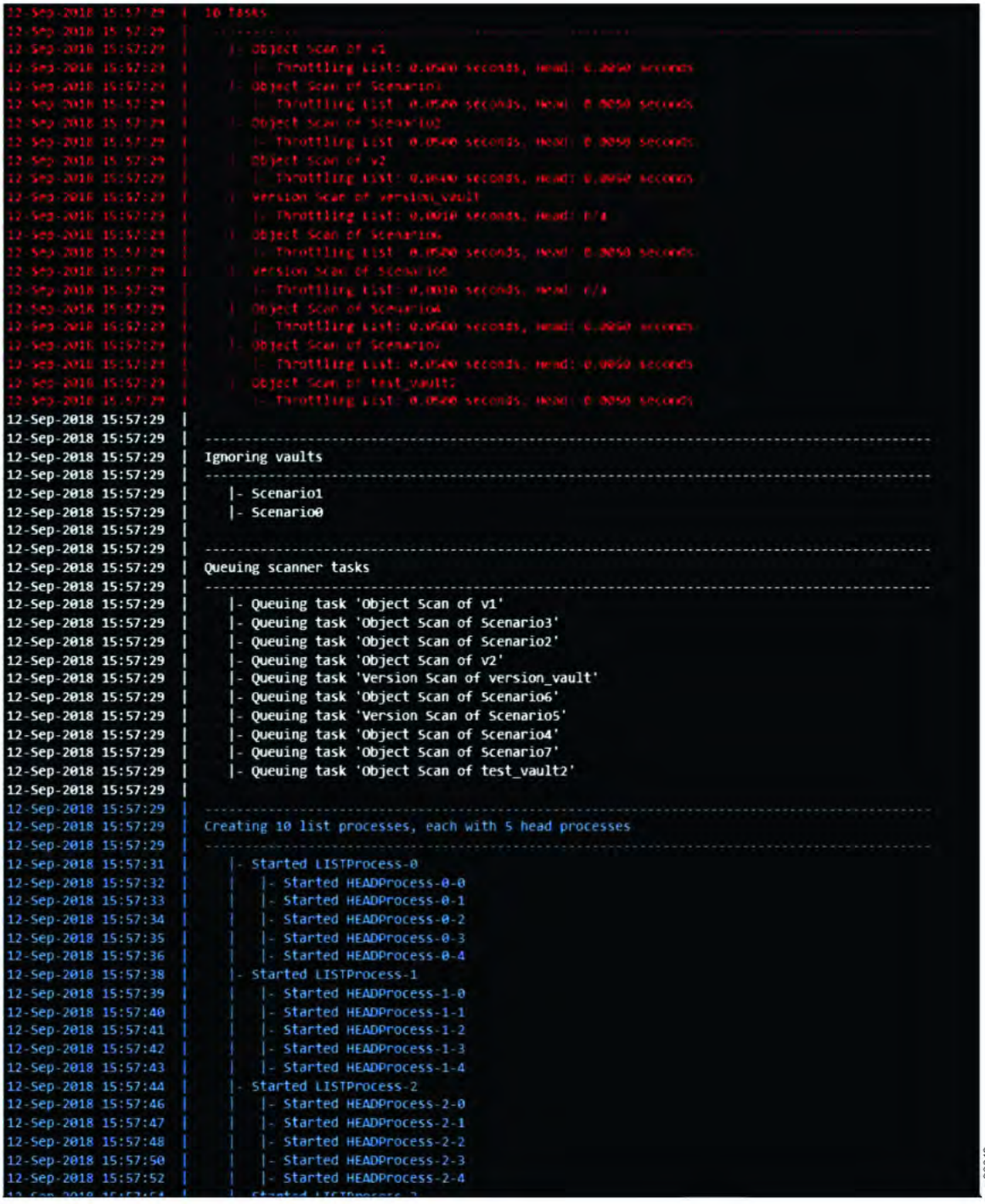
pov00011

Figure 4. Scanner debug (continued)

```
12-Sep-2018 15:57:25 |
12-Sep-2018 15:57:25 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:25 | Python package setup
12-Sep-2018 15:57:25 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:25 | """Setup"""
12-Sep-2018 15:57:25 | from setuptools import setup, find_packages
12-Sep-2018 15:57:25 | setup(
12-Sep-2018 15:57:25 |     name='ibm_cos_scanner',
12-Sep-2018 15:57:25 |     version='2.0.0',
12-Sep-2018 15:57:25 |     packages=find_packages(),
12-Sep-2018 15:57:25 |     include_package_data=True,
12-Sep-2018 15:57:25 |     zip_safe=True,
12-Sep-2018 15:57:25 |     url='www.ibm.com',
12-Sep-2018 15:57:25 |     license='See LICENSE folder',
12-Sep-2018 15:57:25 |     author=IBM',
12-Sep-2018 15:57:25 |     description='IBM COS Scanner / Spectrum Discover Notifier'
12-Sep-2018 15:57:25 | )
12-Sep-2018 15:57:25 |
12-Sep-2018 15:57:25 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:25 | Initialising IBM COS Scanner. Reading config
12-Sep-2018 15:57:25 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:25 | Retrieving System Advanced Configuration
12-Sep-2018 15:57:25 | Calling https://172.19.17.38/manager/api/json/1.0/viewSystemConfiguration.adm
12-Sep-2018 15:57:26 |    |- OK
12-Sep-2018 15:57:26 |    |- dsNet Name: est
12-Sep-2018 15:57:26 |    |- dsNet UUID: f7d033c2-9066-499a-a883-829860d4d865
12-Sep-2018 15:57:26 |
12-Sep-2018 15:57:26 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:26 | Retrieving user's access keys
12-Sep-2018 15:57:26 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:26 | Calling https://172.19.17.38/manager/api/json/1.0/listMyAccessKeys.adm
12-Sep-2018 15:57:26 |    |- OK
12-Sep-2018 15:57:26 |    |- Accesser credentials successfully retrieved from Manager API
12-Sep-2018 15:57:26 |
12-Sep-2018 15:57:26 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:26 | Retrieving vault information from dsNet
12-Sep-2018 15:57:26 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:26 | Calling https://172.19.17.38/manager/api/json/1.0/viewSystemConfiguration.adm
12-Sep-2018 15:57:27 |    |- OK
12-Sep-2018 15:57:27 |
12-Sep-2018 15:57:27 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:27 | Retrieving vault size information from dsNet
12-Sep-2018 15:57:27 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:27 | Calling https://172.19.17.38/manager/api/json/1.0/listVaults.adm
12-Sep-2018 15:57:27 |    |- OK
12-Sep-2018 15:57:27 |
12-Sep-2018 15:57:27 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:27 | Retrieving dsNet device information
12-Sep-2018 15:57:27 | ------------------------------------------------------------------------------------------
12-Sep-2018 15:57:27 | Calling https://172.19.17.38/manager/api/json/1.0/listDevices.adm
12-Sep-2018 15:57:28 |    |- OK
12-Sep-2018 15:57:28 | 3 devices found
12-Sep-2018 15:57:28 |    |- device_id:1    manager     172.19.17.38   0e547a7f-849a-79e8-102c-2026af755443vm-eqx20420l-
mgr-03
12-Sep-2018 15:57:28 |    |- device_id:2    accesser    172.19.17.39   39a14ec8-090c-79d8-10f9-255249197f43vm-eqx20420l-
```

pov00012

Figure 5. Scanner debug (continued)



Configure IBM Cloud Object Storage notifications for IBM Data Cataloging

Ingesting IBM Cloud® Object Storage event records into IBM Data Cataloging requires the user to enable the Notification service on the IBM Cloud Object Storage system. Thereafter, the user must connect the IBM Cloud Object Storage system to the IBM Cloud Object Storage connector Kafka topic on the IBM Data Cataloging cluster. The name of this connector topic is **`cos-le-connector-topic`**.

A combination of SASL and TLS is used to authenticate and encrypt the connection between the IBM Cloud Object Storage source system and the Kafka brokers which reside in the IBM Data Cataloging cluster. The certificate and credentials required to establish this connection might be obtained directly from the IBM Data Cataloging cluster by the IBM Data Cataloging storage administrator.

For information on how to enable and configure the IBM Cloud Object Storage Notification service with the IBM Data Cataloging provided credentials, see IBM Cloud Object Storage Administration System documentation.

The following information is required to establish a secure connection between IBM Cloud Object Storage and IBM Data Cataloging:

Hosts

One or more of the IBM Data Cataloging Kafka brokers is in the format: *host1:port,host2:port*. The Kafka producers on the IBM Cloud Object Storage system will retrieve the full list of IBM Data Cataloging Kafka brokers from the first host that is alive and responding. The broker's host and port (the list configured might contain more than one broker) for SASL SSL can be obtained by the IBM Data Cataloging storage administrator from the following location on the IBM Data Cataloging master node: /etc/kafka/server.properties.

Authentication credentials

The username is `cos` and the password can be obtained by the IBM Data Cataloging storage administrator from the following location on the IBM Data Cataloging master node: /etc/kafka/sasl_password.

Certificate PEM for TLS encryption

This the CA certificate that is used to sign the Kafka server and client certificates for the IBM Data Cataloging cluster. It might be obtained by the IBM Data Cataloging storage administrator from the following location on the IBM Data Cataloging master node: /etc/kafka/ca.crt.
This file is in the PEM format and the entirety of its contents must be pasted into the Certificate PEM field of the COS Notifications configuration panel.

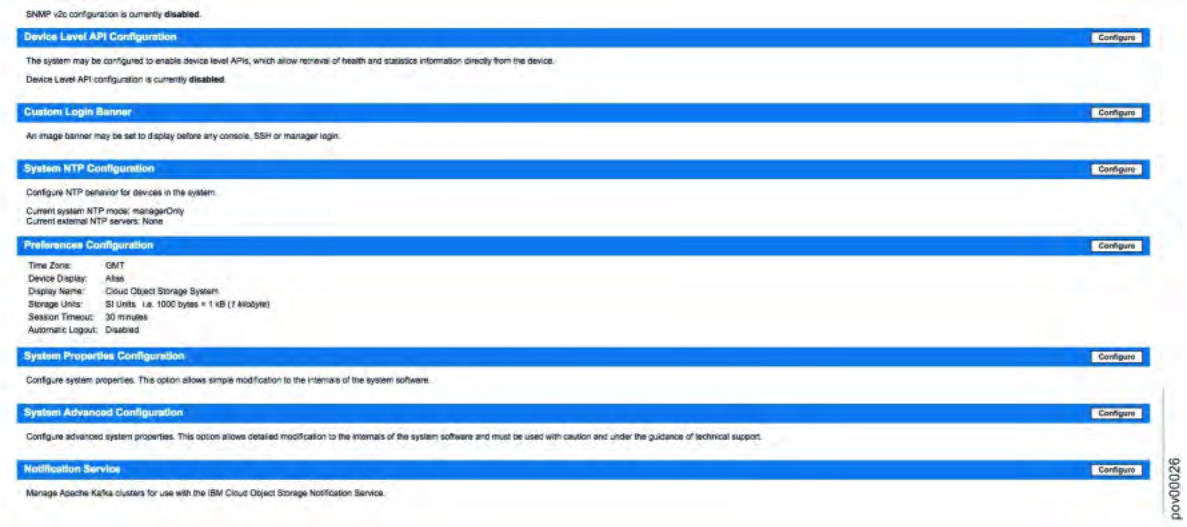
Enabling IBM Cloud Object Storage notification services

The IBM Cloud® Object Storage notification service can be enabled with the information that follows:

Procedure

1. Log in to the IBM Cloud Object Storage Manager Admin console (https://manager_host/manager/login.adm) with a username of **admin** and a password of **password**.
   If you defined your own password, use your pre-defined password. If you do not have a pre-defined password use the default password.
2. Select the Administration tab.
3. Scroll to the end of the page and select Configure the Notification Service.

Figure 1. Configurations



Before you add a notification service to the IBM Cloud Object Storage platform, you must obtain some information from the IBM Data Cataloging server.

To authenticate the IBM Cloud Object Storage notification service, you can capture the Kafka user name and password from the files on the IBM Data Cataloging platform.

If the Transport Layer Security (TLS) is enabled in the IBM Cloud Object Storage notification service, you can also copy the certificate authority (CA) in PEM format from the IBM Data Cataloging platform. After you collect the information, you can add the information to the notification service configuration.

- **Authenticating, encrypting, and enabling**
  Configure Kafka authentication, TLS encryption, and notification service settings by extracting certificates and credentials from the IBM Spectrum Discover server configuration files.
- **Authenticating, encrypting, and enabling services on Red Hat OpenShift**
  Follow the procedure shown to authenticate, encrypt, and enable services on Red Hat® OpenShift®:

Authenticating, encrypting, and enabling

Configure Kafka authentication, TLS encryption, and notification service settings by extracting certificates and credentials from the IBM Spectrum Discover server configuration files.

- Log in to the IBM Data Cataloging server and extract the information from the following example, which contains an example of Kafka user name and password.

```
moadmin@server kafka]$ cd /etc/kafka

[moadmin@server kafka]$ cat kafka_server_jaas.conf
KafkaServer {
org.apache.kafka.common.security.plain.PlainLoginModule required
user_cos="meezDMxFNZJMSxdyWQKSjVbs";
};
```

```
User= cos
Password = meezDMxFNZJMSxdyWQKSjVbs
```

Encryption

The following information shows an example of a certificate of authority for the PEM file.

1. Log in to the IBM Data Cataloging server as moadmin.
2. In the /etc/kafka/ca.crt file, copy the block of text that starts with **BEGIN CERTIFICATE** and that ends with **END CERTIFICATE**. The following example displays what the copied block of text might look like:

```
-----BEGIN CERTIFICATE-----
MIIExTCCA62gAwIBAgIJAKMX/n6ULb6YMA0GCSqGSIb3DQEBCwUAMIGYMQswCQYD
VQQGEwJHQjEOMAwGA1UECAwFSEFOVFMxEDAOBgNVBAcMB0h1cnNsZXkxDDAKBgNV
BAoMA0lCTTEZMBcGA1UECwwQc3BlY3RydW1kaXNjb3ZlcjEZMBcGA1UEAwwQc3Bl
Y3RydW1kaXNjb3ZlcjEjMCEGCSqGSIb3DQEJARYUbWxhd3JlbmNlQHVrLmlibS5j
b20wHhcNMTkwMTAyMTY1MDU5WhcNMzgxMjI4MTY1MDU5WjCBmDELMAkGA1UEBhMC
R0IxDjAMBgNVBAgMBUhBTlRTMRAwDgYDVQQHDAdIdXJzbGV5MQwwCgYDVQQKDANJ
Qk0xGTAXBgNVBAsMEHNwZWN0cnVtZGlzY292ZXIxGTAXBgNVBAMMEHNwZWN0cnVt
ZGlzY292ZXIxIzAhBgkqhkiG9w0BCQEWFG1sYXdyZW5jZUB1ay5pYm0uY29tMIIB
IjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwg7z4gDeWlkeJjPvj3wobDBB
JrHJngooDbPLicRSf/yjl1NgwbWbjIjIeL9R8My+24hRUGfym9IWCM8qMWyEHG+w
+Rr/6jdQyD89j+m1c2ly3nDhXYsTQZR03UylC/TimF6fc07CfuQ1E2ljHf/JXVK4
ESVilhZR23/tWIfbITZmLvdftJSx0Kgu0Ow4BIr9kpQ3bXwt/eoDvAhdKztDouWN
lYCGmdzFOi6E3asspxHhcsGW3bcMu5mqzT6BEnSzrxr8kRbRDL6Q0Pqv33XVxP6z
OHIvv1uFg9Vq6XHIZLBhWNDqPgYoAbT0Q43vUxk7mJ3uJQY6bgbfuEa+PxygQwID
AQABo4IBDjCCAQowHQYDVR0OBBYEFEKxmmHeSfxgHuFL1dd82WMyf190MIHNBgNV
HSMEgcUwgcKAFEKxmmHeSfxgHuFL1dd82WMyf190oYGepIGbMIGYMQswCQYDVQQG
EwJHQjEOMAwGA1UECAwFSEFOVFMxEDAOBgNVBAcMB0h1cnNsZXkxDDAKBgNVBAoM
A0lCTTEZMBcGA1UECwwQc3BlY3RydW1kaXNjb3ZlcjEZMBcGA1UEAwwQc3BlY3Ry
dW1kaXNjb3ZlcjEjMCEGCSqGSIb3DQEJARYUbWxhd3JlbmNlQHVrLmlibS5jb22C
CQCjF/5+lC2+mDAMBgNVHRMEBTADAQH/MAsGA1UdDwQEAwIBBjANBgkqhkiG9w0B
AQsFAAOCAQEANINRvyeuJh69iRK5dPJssmcISXcZv4X33ukAyRt4zLNFToSkTfj2
ZAtQCNgQNl9Ln7Twuit+e6wifxAkA+UD7wrxMzb32+Mpw/XNzo5DnhInfvkAfC62
SHqWIaqTLXDeGbE8O7ieFsI7kAgEQCf23z/vESB2+m1XBI1UcuxMioYwX4YTb14/
GLDJkqhXMLWV+h/7NU7KbERSBia24N5zlR6Ed/rx83uD2AwBnBqt24sD6Q8Gbm+e
HLMv0JrH1vty1vGsfkZnSHb+E6V/5+GsnpIaDyIpsCvM1LqS/wMzBg9hlT5sii8l
mmqMTK6yqcqS7CfWFv/DjQr/i9ECyJ8fAQ==
-----END CERTIFICATE-----
```

Notification service configuration setup

1. Check Enable Configuration.

```
NAME: <NAME>
Topic: cos-le-connector-topic
Hosts: <SD hostname> :9093
Type: IBM Spectrum Discover
```

Enabling authentication

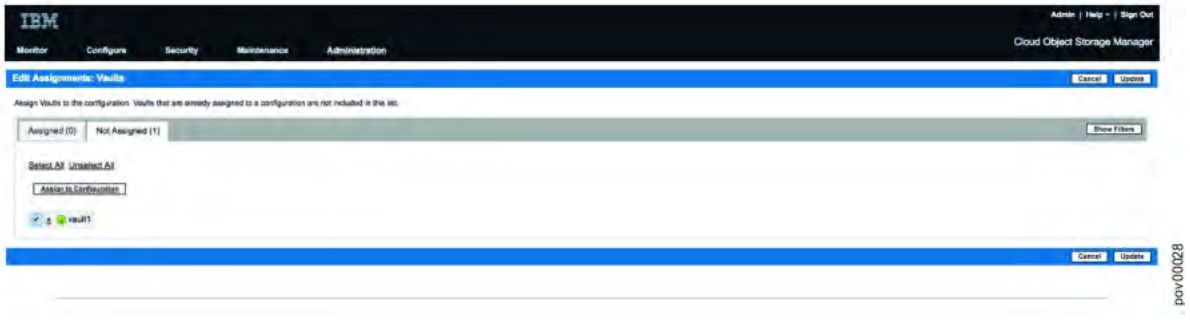
- Check Enable authentication.

```
Username: cos
Password: <PASSWORD>
```

Enabling encryption

1. Check Enable TLS for Apache Kafka network connections.
2. Add the certificate PEM file from the IBM Data Cataloging platform. See Figure 1.

Figure 1. Add a storage vault to the configuration



Authenticating, encrypting, and enabling services on Red Hat OpenShift

Follow the procedure shown to authenticate, encrypt, and enable services on Red Hat® OpenShift®:

- Log in to IBM Data Cataloging server and issue the following command to extract the password information:

```
oc get secret kafka-sasl-password -n ibm-data-cataloging -o jsonpath='{.data.password}'
```

Encryption

The following information shows an example of a certificate of authority for the PEM file:

1. Log in to the IBM Data Cataloging server as moadmin.
2. Issue the following command to extract the information:

```
oc get secret kafka -n ibm-data-cataloging -o jsonpath='{.data.sasl_ca\.crt}' | base64 -d
```

3. Copy the block of text that starts with `BEGIN CERTIFICATE` and ends with `END CERTIFICATE`. The following example displays what the copied block of text might look like:

```
-----BEGIN CERTIFICATE-----
MIIExTCCA62gAwIBAgIJAKMX/n6ULb6YMA0GCSqGSIb3DQEBCwUAMIGYMQswCQYD
VQQGEwJHQjEOMAwGA1UECAwFSEFOVFMxEDAOBgNVBAcMB0h1cnNsZXkxDDAKBgNV
BAoMA01CTTEZMBcGA1UECwwQc3Bl Y3RydW1kaXNjb3ZlcjEZMBcGA1UEAwwQc3Bl
Y3RydW1kaXNjb3ZlcjEjMCEGCSqGSIb3DQEJARYUbWxhd3JlbmNlQHVrLmlibS5j
b20wHhcNMTkwMTAyMTY1MDU5WhcNMzgxMjI4MTY1MDU5WjCBmDELMAkGA1UEBhMC
R0IxDjAMBgNVBAgMBUhBTlRTMRAwDgYDVQQHDAdIdXJzbGV5MQwwCgYDVQQKDANJ
Qk0xGTAXBgNVBAsMEHNwZWN0cnVtZGlzY292ZXIxGTAXBgNVBAMMEHNwZWN0cnVt
ZGlzY292ZXIxIzAhBgkqhkiG9w0BCQEWFG1sYXdyZW5jZUB1ay5pYm0uY29tMIIB
IjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwg7z4gDeWlkeJjPvj3wobDBB
JrHJngooDbPLicRSf/yjl1NgwbWbjIjIeL9R8My+24hRUGfym9IWCM8qMWyEHG+w
+Rr/6jdQyD89j+m1c2ly3nDhXYsTQZR03UylC/TimF6fc07CfuQ1E2ljHf/JXVK4
ESVilhZR23/tWIfbITZmLvdftJSx0Kgu0Ow4BIr9kpQ3bXwt/eoDvAhdKztDouWN
lYCGmdzFOi6E3asspxHhcsGW3bcMu5mqzT6BEnSzrxr8kRbRDL6Q0Pqv33XVxP6z
OHIvv1uFg9Vq6XHIZLBhWNDqPgYoAbT0Q43vUxk7mJ3uJQY6bgbfuEa+PxygQwID
AQABo4IBDjCCAQowHQYDVR0OBBYEFEKxmmHeSfxgHuFL1dd82WMyf190MIHNBgNV
HSMEgcUwgcKAFEKxmmHeSfxgHuFL1dd82WMyf190oYGepIGbMIGYMQswCQYDVQQG
EwJHQjEOMAwGA1UECAwFSEFOVFMxEDAOBgNVBAcMB0h1cnNsZXkxDDAKBgNVBAoM
A01CTTEZMBcGA1UECwwQc3Bl Y3RydW1kaXNjb3ZlcjEZMBcGA1UEAwwQc3Bl Y3Ry
dW1kaXNjb3ZlcjEjMCEGCSqGSIb3DQEJARYUbWxhd3JlbmNlQHVrLmlibS5jb22C
CQCjF/5+lC2+mDAMBgNVHRMEBTADAQH/MAsGA1UdDwQEAwIBBjANBgkqhkiG9w0B
AQsFAAOCAQEANINRvyeuJh69iRK5dPJssmcISXcZv4X33ukAyRt4zLNFToSkTfj2
ZAtQCNgQNl9Ln7Twuit+e6wifxAkA+UD7wrxMzb32+Mpw/XNzo5DnhInfvkAfC62
SHqWIaqTLXDeGbE8O7ieFsI7kAgEQCf23z/vESB2+mlXBI1UcuxMioYwX4YTb14/
GLDJkqhXMLWV+h/7NU7KbERSBia24N5zlR6Ed/rx83uD2AwBnBqt24sD6Q8Gbm+e
HLMv0JrH1vty1vGsfkZnSHb+E6V/5+GsnpIaDyIpsCvM1LqS/wMzBg9hlT5sii8l
mmqMTK6yqcqS7CfWFv/DjQr/i9ECyJ8fAQ==
-----END CERTIFICATE-----
```

Notification service configuration setup

- Check Enable Configuration.

```
NAME: <NAME>
Topic: cos-le-connector-topic
Hosts: <SD ipaddress> :443
Type: IBM Spectrum Discover
```

Enabling authentication

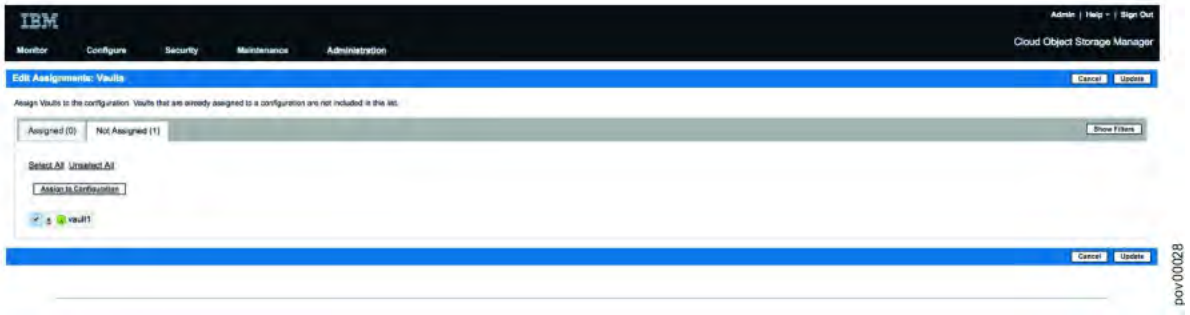
- Check Enable authentication.

```
Username: cos
Password: <PASSWORD>
```

Enabling encryption

1. Check Enable TLS for Apache Kafka network connections.
2. Add the certificate PEM file from the IBM Data Cataloging platform. See Figure 1.

Figure 1. Add a storage vault to the configuration



Testing the IBM Cloud Object Storage notification service

To test the IBM Cloud® Object Storage notification service, the tester can populate the IBM Cloud Object Storage vault with test data.

About this task

You can use a number of methods to write files to an IBM Cloud Object Storage vault, but you can use cURL directly on IBM Data Cataloging platform. cURL is a computer software project that provides a library and command-line tool for transferring data that uses various protocols.

Procedure

1. Create a test file, for example, object_1.txt.
   The test file can be any file that contains data.
2. Write a file to the IBM Cloud Object Storage vault by using cURL.
   Requirements

   IBM® COS Vault Name (vault1) [anonymous access enabled]
   IBM COS Accesser IP address

Example

- curl -X PUT -i -T object_1.txt http://9.11.200.208/vault1/object_1.txt
- HTTP/1.1 100 ContinueHTTP/1.1 200 OK
- Date: Fri, 04 Jan 2019 13:21:14 Greenwich mean time
- X-Clv-Request-Id: a9ad657a-a919-4b13-9b72-961ae8c57e3c
- Server: 3.14.0.23
- X-Clv-S3-2.5
- x-amz-request-id: a9ad657a-a919-4b13-9b72-961ae8c57e3c
- ETag: "7c517c7108f7180377e7b37db2e39261"
- Content-Length: 0

- **Monitoring the IBM Cloud Object Storage accesser logs**
  To determine whether a file is successfully written to the IBM Cloud Object Storage vault and a notification is successfully sent to the IBM Data Cataloging server, the accesser logs can be monitored on the IBM Cloud Object Storage Accesser server.
- **Monitoring the IBM Data Cataloging producer IBM Cloud Object Storage logs**
  When the IBM Data Cataloging server receives a notification from the IBM Cloud Object Storage platform, the IBM Data Cataloging producer IBM Cloud Object Storage records a transaction.
- **Monitoring the IBM Data Cataloging dashboard for IBM Cloud Object Storage ingestion**
  You can monitor the IBM Data Cataloging dashboard for IBM Cloud Object Storage ingestion.

Monitoring the IBM Cloud Object Storage accesser logs

To determine whether a file is successfully written to the IBM Cloud® Object Storage vault and a notification is successfully sent to the IBM Data Cataloging server, the accesser logs can be monitored on the IBM Cloud Object Storage Accesser server.

In the following example, an object that is written to vault1 results in the sending of one notification to the IBM Data Cataloging server. The user must have access privileges to log on to the IBM Cloud Object Storage Accesser host to check the log files.

Confirm that an object is stored in the IBM Cloud Object Storage vault.

```
root@ibm_accesser:/var/log/dsnet-core# tail -f http.log
9.11.201.78 - ""  -  [04/Jan/2019:13:21:14 +0000] "PUT /vault1/object_1.txt HTTP/1.1" 200 0 "-" "curl/7.29.0" 22
```

Confirm that a notification is sent to the IBM Data Cataloging server.

```
root@ibm_accesser:/var/log/dsnet-core# tail -f notification.log
{"time":"2019-01-04T13:21:14.668Z","request_id":"a9ad657a-a919-4b13-9b72-961ae8c57e3c","retried":true,"success":true,
"request_time":"2019-01-04T13:21:14.567Z","kafka_config_uuid":"d842c7a0-9c36-412e-8908-8ad5120a261e","topic":"cos-le-connector-
topic"
```

Monitoring the IBM Data Cataloging producer IBM Cloud Object Storage logs

When the IBM Data Cataloging server receives a notification from the IBM Cloud® Object Storage platform, the IBM Data Cataloging producer IBM Cloud Object Storage records a transaction.

A successful notification is recorded as an offset value of one, when a notification is received from IBM Cloud Object Storage platform.

```
[moadmin@data_cataloging]$ oc logs -f -n producercos  kindled-alligator-producer-cos-producer-9f6966b4-8jsg7
break time. waiting for work...
2019-01-04 13:21:19.187 > offset_commit_cb: success, offsets:[{part: 0, offset: 1, err: none}]
```

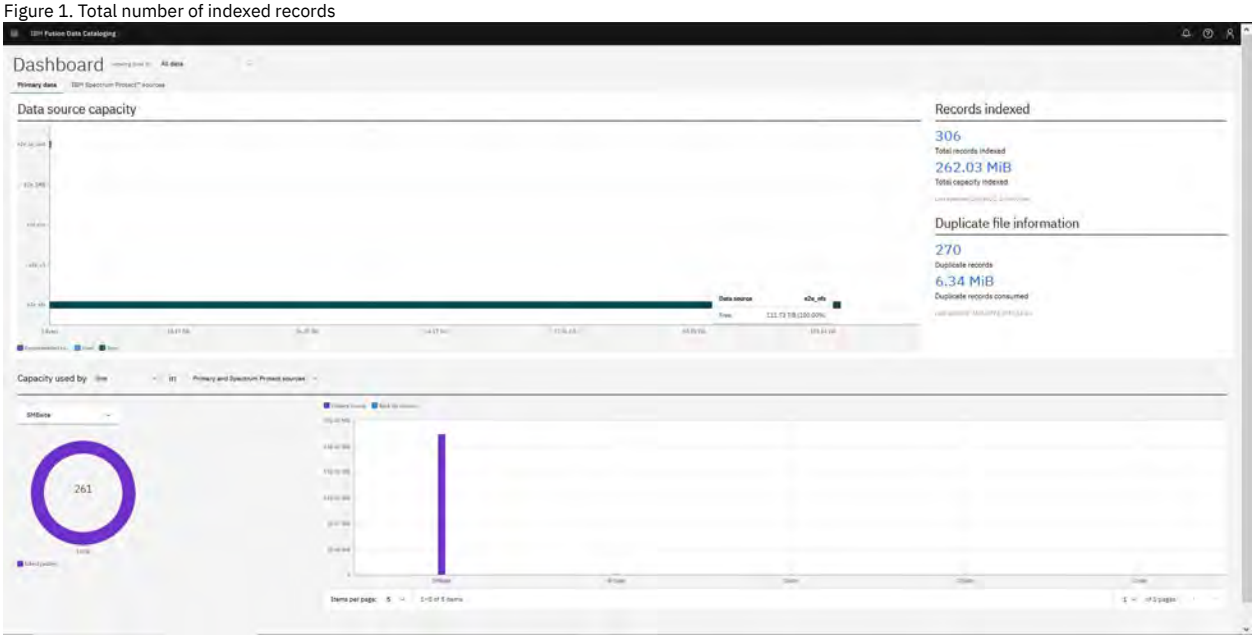
Monitoring the IBM Data Cataloging dashboard for IBM Cloud Object Storage ingestion

You can monitor the IBM Data Cataloging dashboard for IBM Cloud® Object Storage ingestion.

After IBM Cloud Object Storage notifications are ingested from the IBM Cloud Object Storage platform, the IBM Data Cataloging dashboard displays the total number of indexed records.

Note: The IBM Data Cataloging dashboard can take approximately 30 minutes to display the total number of indexed records.
See Figure 1.

Figure 1. Total number of indexed records



IBM Data Cataloging and S3 object storage data source connections

Use this information to understand how IBM Data Cataloging works with S3-compliant object store.

- **Creating an S3 object storage data connection**
  Use this information to create a connection to an S3-compliant object store.
- **Scanning an S3 object storage data connection**
  Use this information to scan a connection for an S3-compliant object store.
- **Best practices for scanning an S3 object storage system**
  Use best practices for scanning S3-compliant object storage systems.

Creating an S3 object storage data connection

Use this information to create a connection to an S3-compliant object store.

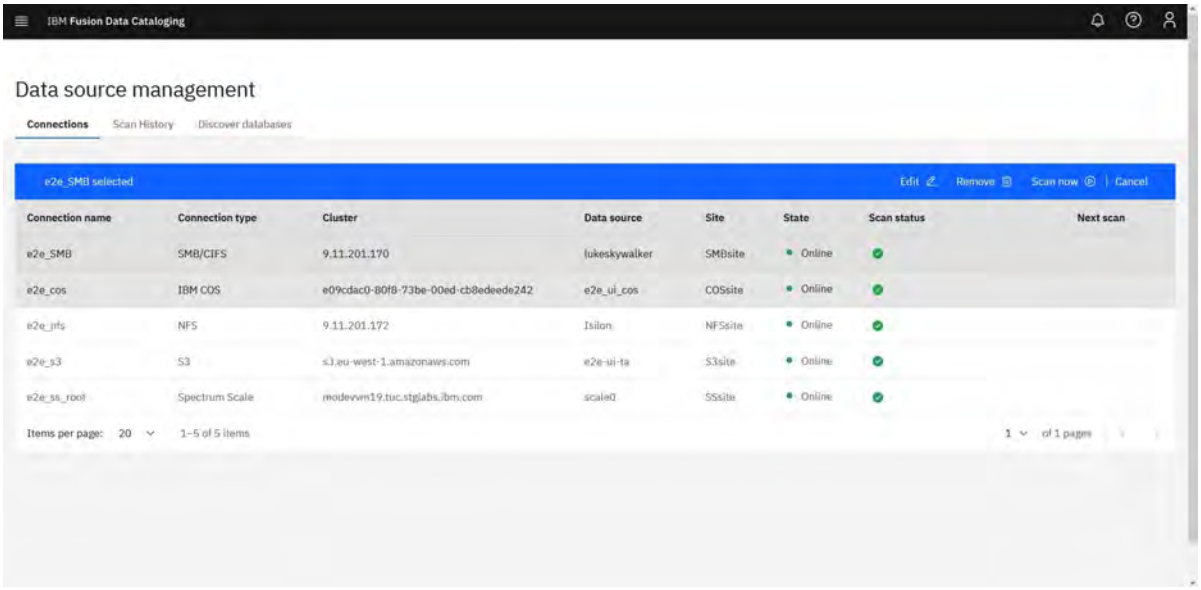
About this task

To create the S3-compliant connection:
Note: The storage host supports SSL by default. If `http` is required, then explicitly add `http://` prior to the hostname or IP address.

Procedure

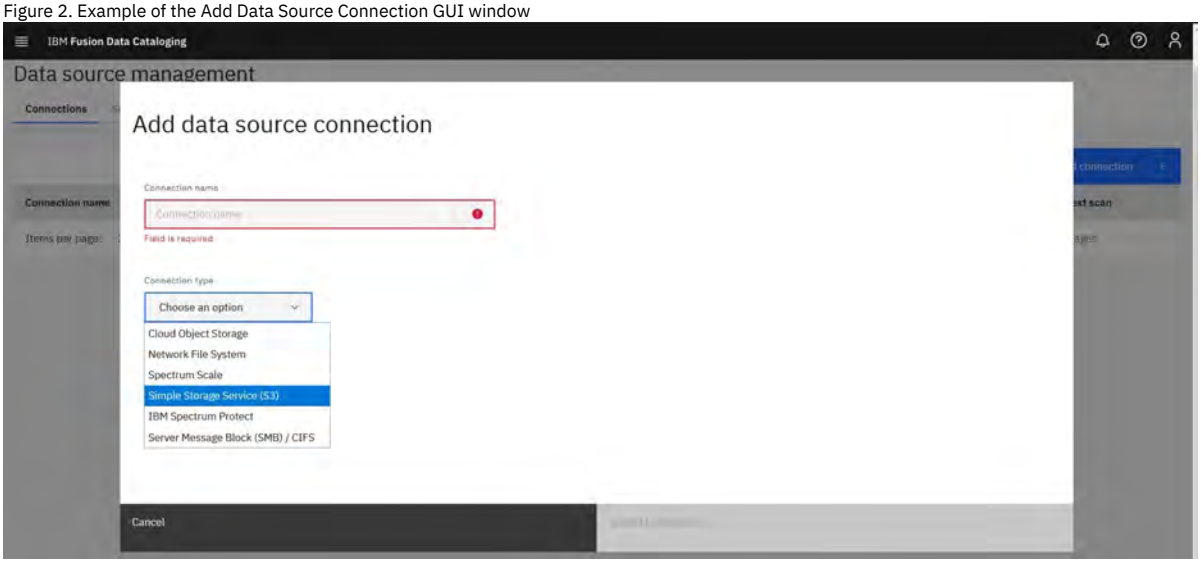
1. Log in to the IBM Data Cataloging graphical user interface (GUI) with a user ID that is associated with data administration role.
   The data administration access role is required for creating connections. For more information about role-based access control, go to Role-based access control.
2. Click  and go to Data connections > Connections.
   Clicking Connections displays the different types of data source connection names, platforms, clusters, data source, size, and Add Connection.

   Figure 1. Displaying the source names for Data Source Connections



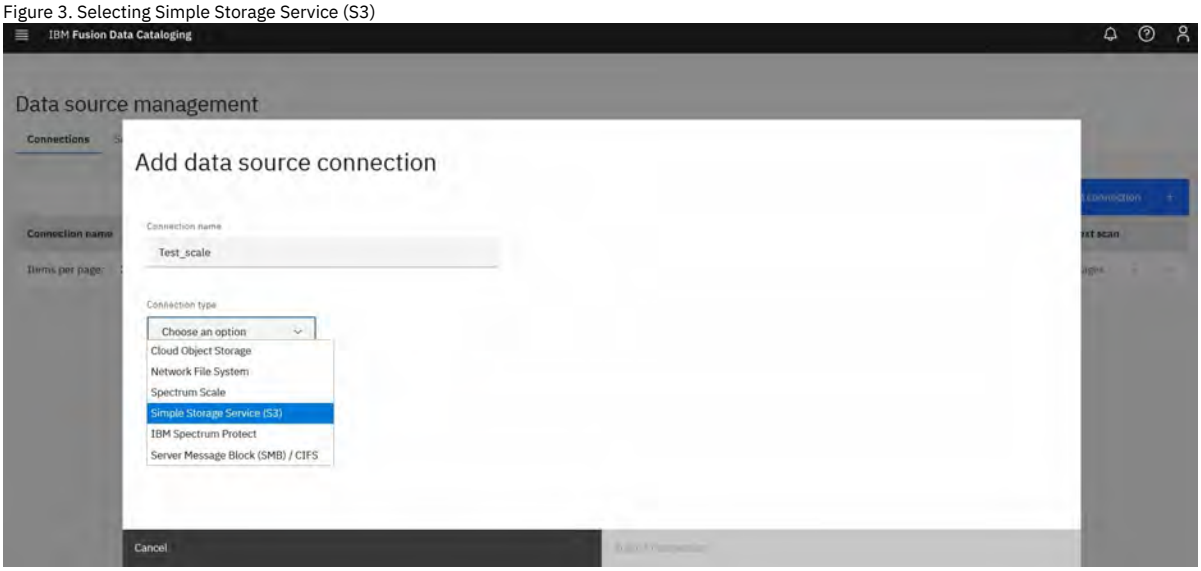
3. Click Add Connection to display a new window that shows Add Data Source Connection.

Figure 2. Example of the Add Data Source Connection GUI window



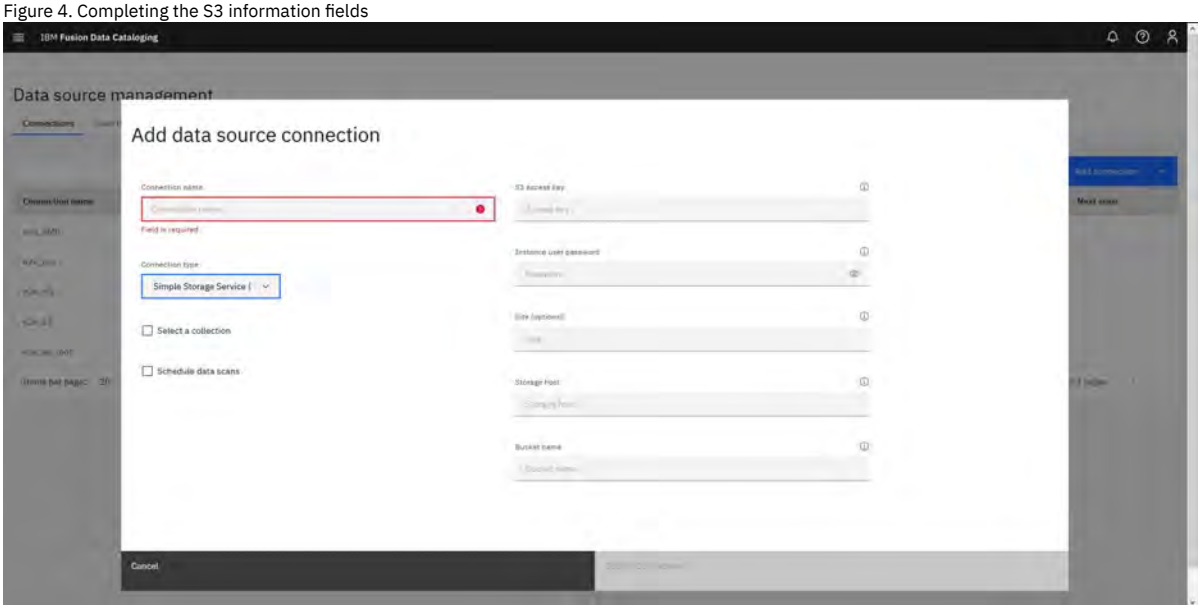
4. Complete the following steps:
   a. In the field for Connection name, define a Connection name.
   b. Choose data source connection type from Connection type list.
5. Select the connection type Simple Storage Service (S3).

Figure 3. Selecting Simple Storage Service (S3)



6. In the screen for S3, complete the fields and then click Submit Connection for the S3 connections manager.

Figure 4. Completing the S3 information fields



S3 access key
: Indicates the access key ID for the S3 object store.

S3 secret key
: Indicates the secret key ID for the S3 object store.

Site
: Indicates the physical location of the records.

Storage host
: Indicates the IP address or host of the storage system.

Bucket name
: Indicates the name of the bucket that you are going to scan.

Scanning an S3 object storage data connection

Use this information to scan a connection for an S3-compliant object store.

About this task

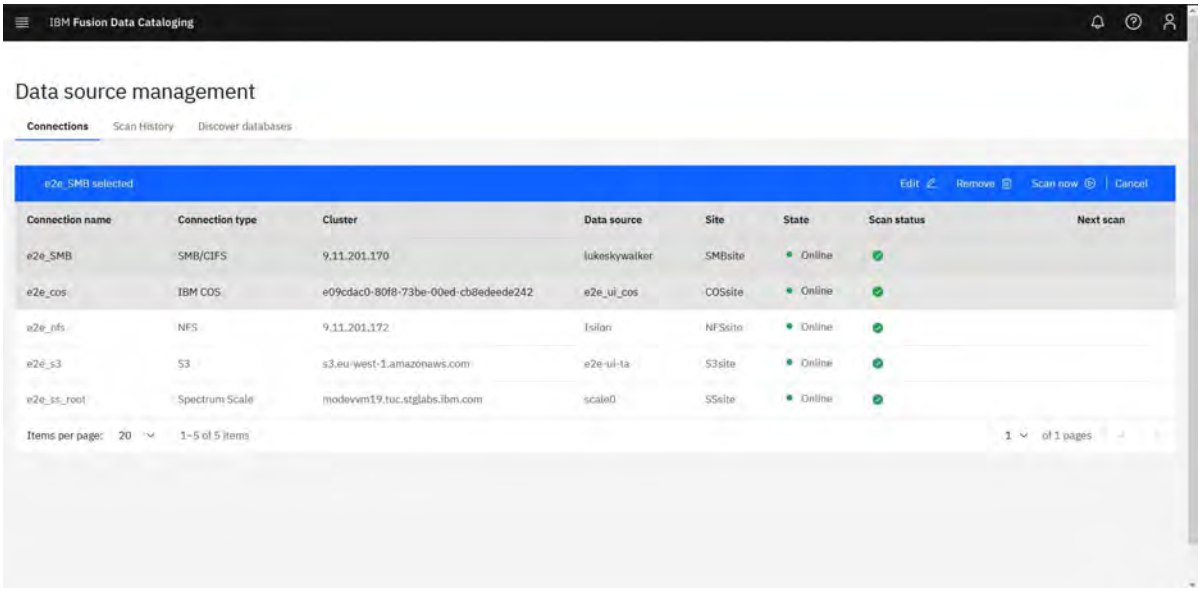
When you initiate a scan from the IBM Data Cataloging graphical user interface (GUI), the metadata is transferred asynchronously back to the IBM Data Cataloging instance.

Automated scanning and data ingestion relies on an established and active network connection between the IBM Data Cataloging instance and the S3 storage source. If the connection cannot be established, the state of the data source connection shows as unavailable, and the option for automated scanning does not appear in the IBM Data Cataloging GUI for that connection.

Procedure

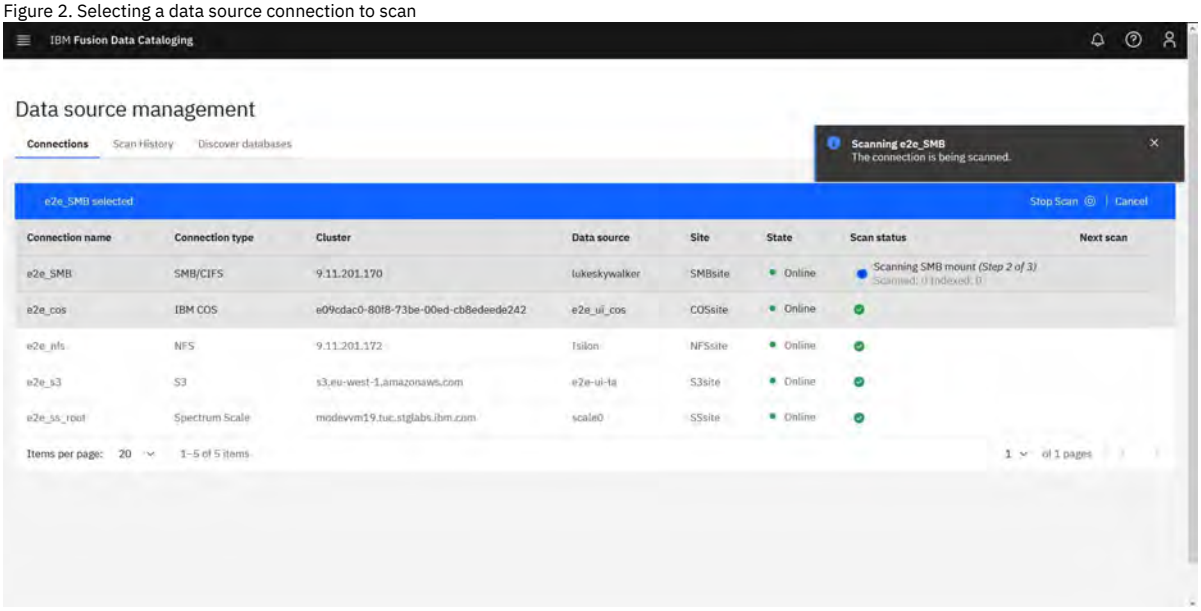
1. Log in to IBM Data Cataloging web interface.
2. Click  go to Data connections > Connections.
   The following figure shows the data connections lists:

Figure 1. Data source connections



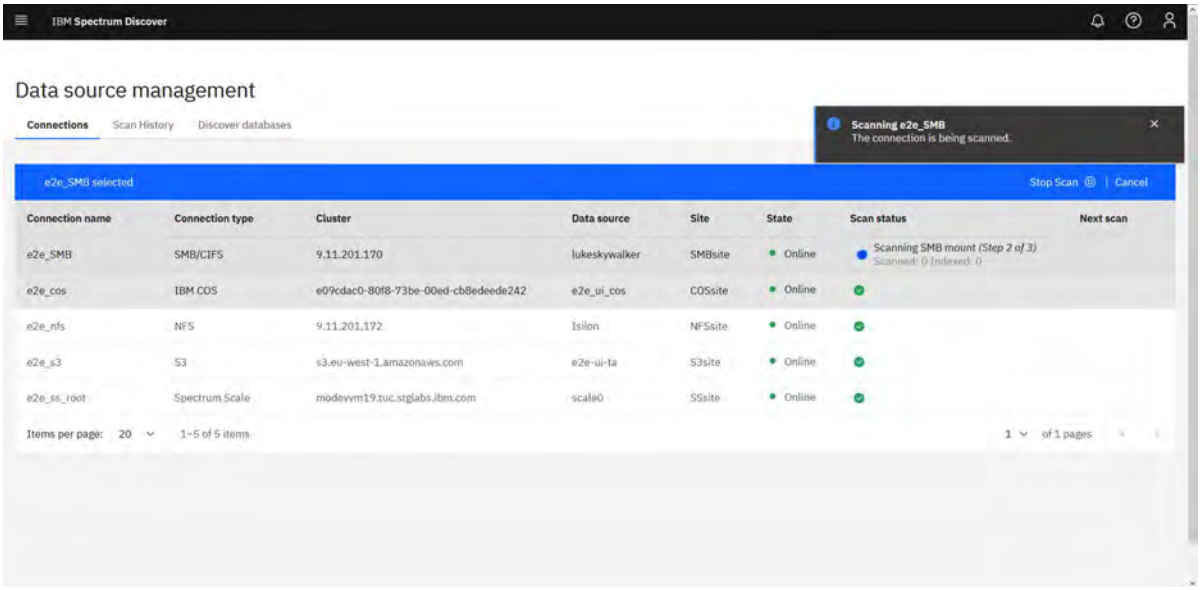
3. Select the data source connection name that you want to scan. Ensure that the **`State`** is listed as **`Online`** to make your system scan ready.
   The following figure shows how to select a data source connection to scan.

Figure 2. Selecting a data source connection to scan



4. Select Scan now to change the status to Scanning.
   The following figure shows an active scan.

Figure 3. Active scans



5. When the scan finishes, the state field returns to a status of Online.

Best practices for scanning an S3 object storage system

Use best practices for scanning S3-compliant object storage systems.

It is recommended to check the log files in the following directories after each scan:

/opt/ibm/metaocean/data/connections/s3/<connection_name>/debug/<scan_timestamp>/scanner.debug indicates whether the scan was successful or not.

/opt/ibm/metaocean/data/connections/s3/<connection_name>/error/<scan_timestamp>/scanner.error contains a list of all the messages that are not delivered to IBM Data Cataloging.

/opt/ibm/metaocean/data/connections/s3/<connection_name>/data/<scan_timestamp>/ contains a subfolder with the scanned data source name. There is a stats folder inside this folder that contains information about the number of objects in the data source or the number of objects or scanned files.

You can also compare the total size of the bucket that is reported in IBM Data Cataloging with the total size of the S3 object store at source (if it is available).

Scanning an Elastic Storage Server data source connection

Use the IBM Data Cataloging GUI to scan an Elastic Storage Server (ESS) data connection.

Procedure

1. Log in to IBM Data Cataloging web interface with data admin privileges.
2. Open an internet browser to the IP address or host name of your IBM Data Cataloging cluster.
   The default credentials to put into the IBM Spectrum Discover dialog box are:

   ```
   Default user: sdadmin
   Default Password: Passw0rd
   ```

3. Click  menu and go to Data connections > Add data source connection.
4. Enter the EMS node as the host system name:

   Connection Name
   : Indicates the connection name, such as `ESS_test`.

   Connection Type
   : Indicates the connection type, which is `Spectrum Scale`.

   User
   : Indicates the user that initiates the scan, such as `root`.

   Password
   : Indicates the password for the user that initiates the scan.

   Working Directory
   : Indicates the working directory for the scan, such as `/gpfd/icp4D_data_fs_master1/sd_scan`.

   Scan Directory
   : Indicates the scan directory, such as `/gpfd/icp4D_data_fs_master1`.

   Site (optional)
   : Indicates the scan site. This field is optional.

   Cluster
   : Indicates the clustered system name that is being scanned, such as `heliumess.tuc.stglabs.ibm.com`.

Host
: Indicates the host system name that is being scanned, such as `9.11.103.45`.

Filesystem
: Indicates the file system name that is being scanned.

Node List
: Indicates the node list of nodes that are being scanned. Enter `gss_ppc64` as the node class to perform the scan.

5. After you complete all fields, select Submit Connection.
6. Select your data source from the table (click the row) and then select Scan Now.
7. You can monitor the status of the scans from Data Connections Overview.
8. You can also search for documents from that data source by using the search navigation icon in the left side of the GUI.
   In the search icon you must:
   a. Enter a query.
   b. Click Search.
   c. Indicate the files that are found in the search results.

Creating a Network File System data source connection

You can use the IBM Data Cataloging graphical user interface to create a Network File System (NFS) data connection.

About this task

Note: For NFS scanning that uses Data ONTAP 8.1.2 or later, export the file system with the following configuration:

Protocol
: NFSv3 and NFSv4

Security type
: UNIX

Client permissions
: A minimum of read-only access is necessary.

Anonymous users
: Root access must be granted.

Setuid and Setgid executable routines are not necessary.
Creating data source connections in IBM Data Cataloging identifies source storage systems that are to be indexed by IBM Data Cataloging.

For some data source types, a network connection can be created to allow for automated scanning and indexing of the source system metadata. IBM Data Cataloging will not index data from unknown sources, so creating a data source connection is the first step towards cataloging any source storage system.

Procedure

1. Log in to the IBM Data Cataloging web interface with a user ID that has the data admin role that is associated with it.
   Note: The data admin access role is necessary for creating connections.
2. Click  menu and go to Data connections to display the connection's source name, platform, cluster, data source, site, next run time, and state of existing data source connections.
3. Click Add connection to display the Add data source connection window.
4. Enter a name in the Connection name field.
5. Select the connection type Network File System from the Connection type list.
6. Complete the fields for the NFS parameters, and click Submit Connection.

Parameters for NFS connections

Connection name
: The name of the connection, an identifier for the user, for example `filesystem1`.
  Note: It must be a unique name within IBM Data Cataloging.

Connection type
: The type of source storage system this connection represents.

Enable NFSv4 ACL scans
: Enables the scan of NFSv4 ACL permission entries.

Data source
: The full name of the data source.

Export path
: The data path from which data is to be exported.

Host
: The host name of the node from which a scan can be initiated.

Site
: The location in which the data source facility is located.

Creating an SMB data source connection

Creating an SMB data connection by using the IBM Data Cataloging graphical user interface.

About this task

Create data source connections in IBM Data Cataloging to identify and index source storage systems.

For some data source types, a network connection is (optionally) created to allow for automated scanning and indexing of the source system metadata. IBM Data Cataloging does not index data from unknown sources, so creating a data source connection is the first step towards cataloging any source storage system.

Procedure

1. Log in to the IBM Data Cataloging web interface with a user ID that is associated with the data admin role.
   Note: The Data admin access role is required for creating connections.
2. Click  menu and go to Data connections > Connections to display the source connection name, platform, cluster, data source, site, next run time, and state of existing data source connections.
3. Click Add connection to display the Add data source connection window.
4. Enter a name in the Connection name field.
5. Select the connection type Server Message Block (SMB) / CIFS from the Connection type list.
6. Complete the fields for the SMB parameters, and click Submit Connection.

Parameters for SMB connections

Connection Name
: Indicates the name of the connection, an identifier for the user, for example Share1.
  Note: It must be a unique name within IBM Data Cataloging.

Share
: Indicates the path name of the SMB/CIFS file share, for example //server/share.

User
: Indicates the user ID that has access permissions to the SMB share, for example DOMAIN1\user1

Password
: Indicates the authentication password for the user ID provided.

Site (Optional)
: Indicates the location in which the data source facility is located.

Creating an IBM Storage Protect data source connection

Creating the IBM Storage Protect data connection by using the IBM Data Cataloging graphical user interface.

About this task

Creating data source connections in IBM Data Cataloging identifies source storage systems that are to be indexed by IBM Data Cataloging.

For some data source types, a network connection is (optionally) created to allow for automated scanning and indexing of the source system metadata. IBM Data Cataloging will not index data from unknown sources, so creating a data source connection is the first step towards cataloging any source storage system.

Procedure

1. Log in to the IBM Data Cataloging web interface with a user ID that has the data admin role that is associated with it.
   Note: The data admin access role is required for creating connections.
2. Click  menu and go to Data connectios > Connections to display the source name, platform, cluster, data source, site, next run time, and state of existing data source connections.
3. Click Add connection to display the Add data source connection window.
4. Enter a name in the Connection name field.
5. Select the connection type IBM Spectrum Protect from the Connection type list.
6. Complete the fields for the IBM Storage Protect connection type, and click Submit Connection.

Parameters for IBM Storage Protect connections

Connection Name
: The name of the connection, an identifier for the user, for example filesystem1.
  Note: It must be a unique name within IBM Data Cataloging.

Spectrum Protect Server IP
: The IP address or host name of the IBM Storage Protect server.

ODBC Port
: The ODBC (open database connector) port for the Protect server (default is 51500).

Instance User
: The name of the IBM Storage Protect database instance user. The default is `tsminst1`.

Instance user password
: The password for the database instance user.

Site (Optional)
: The location in which the data source facility is located.

Configuring data source connections offline

Multiple source data connections can be added to the IBM Data Cataloging system through offline mode.

Before you begin

- You must have the **SD_USER,SD_PASSWORD & OVA** environment variables set in the IBM Data Cataloging system. To verify the details, enter `gettoken`.
- Enter the connection values inside the JSON file and add it to IBM Data Cataloging host by using **load_connection <nameofjsonfile>**.
  A sample JSON file with connection values as shown.

```
{
 "name": "SpectrumScale2Million",
 "platform": "Spectrum Scale",
 "cluster": "ctolib.cluster1",
 "datasource": "scale0",
 "site":"Offline Scale"
}
```

Procedure

1. Log in to the IBM Data Cataloging host system with following credentials.

   ```
   Host: <IBM Data Cataloging_hostname>
   login: moadmin
   Password: Passw0rd
   ```

2. SCP the list.metaocean file to the IBM Data Cataloging VM by using the following command before you run the ingestion script.

   ```
   sudo scp moadmin@modevdump:/mnt/datadump/scans/<file_name>/home/moadmin/
   ```

3. After the list.metaocean file is copied to the IBM Data Cataloging host, issue the following command to ingest the file.

   ```
   $ ingest <file_name>
   ```

4. To monitor the ingestion process by following the producer log, issue the following commands.

   ```
   $ kp | grep spectrum-discover-producer-<connection_name>-scan
   ```

   Note: The preceding command returns the producer pod log.

   ```
   $ oc logs -f -n ibm-data-cataloging spectrum-discover-producer-<connection_name>-scan-<randomchars>
   ```

Editing and using the TimeSinceAccess and Size Range buckets

Users can group or aggregate data into three user-defined bucket ranges. The three user-defined bucket ranges are TimeSinceAccess, Size Range and FileGroup.

The TimeSinceAccess bucket groups files and objects based on the time they were last accessed. The SizeRange bucket groups files and objects based on their size. The File Group bucket groups files based on their file type or extension. All three buckets can be customized to better align with the user's requirements. Both the TimeSinceAccess and the SizeRange buckets have up to five custom ranges with user-defined labels.

Note: The **FileGroup** bucket cannot be edited through the user interface and must be modified using REST APIs. For more information, see/db2whrest/v1/buckets/<bucket>: PUT. For more information, see the topic */db2whrest/v1/buckets/<bucket>: PUT* in the *IBM Data Cataloging: REST API Guide*
To access the SizeRange bucket groups, select Metadata > Tags > edit icon for the SizeRange tag. For example, SizeRange can be broken up into 'T-shirt size' ranges where the ranges and labels are:

Table 1. Examples of size ranges and sizes of buckets with user-defined labels

Size range	Size
0 - 4 K	XS
4 K - 1 M	S
1 M - 1 G	M
1 G - 1T B	L
1 TB+	XL

After you change or update a bucket definition, IBM Data Cataloging summarizes the current set of files and objects into their respective bucket ranges. The changes are updated periodically every half an hour; thus, it may take a half an hour or more before the changes are reflected in the Spectrum Discover GUI.
Note: Ensure that the maximum value for each bucket is greater than the value assigned to the previous bucket.
See the Figure 1

Figure 1. Example of how to define the settings for a SizeRange bucket



To open the menu for the TimeSinceAccess buckets select Metadata > Tags > edit icon for the TimeSinceAccess tag. See the Figure 1 for an example.

Figure 2 shows an example of how to modify and define the settings of a bucket that is older than one year.

Figure 2. Example of how to modify and define the settings of a bucket that is older than one year old



IBM Spectrum Storage software requirements

Use IBM Data Cataloging to index metadata from other applications and to orchestrate the data management.

IBM Cloud Object Storage

IBM Data Cataloging indexes metadata from IBM Cloud® Object Storage by receiving notifications that contain metadata from IBM Cloud Object Storage. IBM Data Cataloging also supports scanning IBM Cloud Object Storage to harvest metadata.

The following table shows the minimum required IBM Cloud Object Storage software version to enable metadata harvesting with IBM Data Cataloging:

Table 1. IBM Cloud Object Storage software requirements

Component	Version
IBM Cloud Object Storage	3.14.0 and higher

IBM Storage Scale

IBM Data Cataloging indexes metadata from IBM Storage Scale by scanning IBM Storage Scale file systems. The IBM Storage Scale watch folders technology preview also enables IBM Storage Scale to send events that contain metadata to IBM Data Cataloging.

The following table lists the minimum required IBM Storage Scale software versions:

Table 2. IBM Spectrum Scale software requirements

Component	Feature	Version
IBM Storage Scale	Scanning	4.2.3.x and higher
IBM Storage Scale	Live events	(Advanced and Data Management Editions, only) 5.0.4.1 and higher
IBM Storage Scale	Data management that uses ScaleAFM Application	5.1 and higher

The following requirements are needed to enable live events:

- You must use IBM Storage Scale 5.0.3.x. Due to an IBM Storage Scale performance issue that might result in the unexpected suspension of the IBM Storage Scale watch, IBM Data Cataloging recommends the use of IBM Storage Scale 5.0.4.x.
- Watch folder must be enabled for the Scale cluster.
- A minimum of three nodes on the IBM Storage Scale cluster are required to act as Kafka brokers.
- The IBM Storage Scale nodes that act as brokers must meet a minimum local space requirement of 20 GB each to successfully enable the watch with a secondary sink.

IBM Storage Protect

IBM Data Cataloging indexes metadata from IBM Storage Protect by scanning IBM Storage Protect file systems.

The following table lists the minimum required IBM Storage Protect software versions to enable metadata harvesting with IBM Data Cataloging:

Table 3. IBM Spectrum Protect software requirements

Component	Version
IBM Storage Protect	7.x and higher

IBM Storage Archive

IBM Data Cataloging supports the advanced tiering function with the ScaleILM application.

The following table lists the minimum IBM Spectrum® Archive software version that is required to control the data placement by IBM Data Cataloging.

Table 4. IBM Storage Archive software requirements

Component	Version
IBM Storage Archive Enterprise Edition (EE)	1.3.0.6 and higher

IBM Data Cataloging installation with alternative VLAN

Install and configure IBM Data Cataloging that uses additional VLAN connected through the IBM Fusion upstream links.

For security reasons, the target VLAN network is not available for the IBM Fusion cluster through the default IBM Fusion public network.

There are several approaches available to access alternate VLANs that do not have routable access through the default IBM Fusion network.

The reminder of this document covers the option of OpenShift® **`NetworkAttachmentDefinitions`** to provide IBM Data Cataloging access to alternative VLAN access. It explains the setup and implementation of **`NetworkAttachmentDefinitions`** and the required modifications to the IBM Data Cataloging definitions.

IBM Fusion VLAN Configuration

1. Add VLAN. For procedure, see Adding VLANs.
2. Add VLAN to link.
3. Check switches to ensure that VLAN is available on bond250 and ports.
4. Check the compute node to ensure that VLAN is added.

IBM Data Cataloging Configuration

1. Log in to the Red Hat® OpenShift Container Platform web console.
2. Go to Networking > NetworkAttachmentDefinitions.
3. Click Project drop-down and select ibm-data-cataloging namespace from the list.
4. Click the NAD that you want and check the Container and VM NAD details.
   For example:

   For container access

   Note: Adding extra VLANs to the IBM Fusion switch link creates an additional bridge interface for each additional VLAN on the worker nodes **`bond0`** interface automatically.

```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
annotations:
k8s.v1.cni.cncf.io/resourceName: bridge.network.kubevirt.io/br4001
name: br4001-c
namespace: ibm-data-cataloging
spec:
config: >-
{“cniVersion”: “0.3.1”, “name”: “br4001”, “type”: “macvlan”, “master”: “br4001”,“mode”: “bridge”, “ipam”: { “type”:
“static” , “routes”: [ {“dst”:“10.140.20.0/24”,“gw”:“10.100.100.1” } ] }}
```

Item	Value	Description
Name	br4001-c	**`Network Attachment Definition`**
`dst`	10.140.20.0	Destination subnet
`gw`	10.100.100.1	Router
Name	br4001	Bridge 4001 (auto generated)

   For VM access

   Note: After attached to any VM, additional configuration on the VM interface must be necessary, including IP and routing information.

```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
 name: br4001
 namespace: ibm-datacataloging
spec:
```

```
  nodeSelector:
    vmtest: "true"
  config: >-
    {"name":"br4001","type":"cnv-bridge","cniVersion":"0.3.1","bridge":"br4001","macspoofchk":true,"ipam":{}}
```

Item	Value	Description
Name	br4001	`Network Attachment Definition`
Bridge	br4001	Bridge interface (auto generated)

Container updates

Important: The container updates might change based on your connection type such as NFS, S3, SMB, and Scale. The following example shows the Scale connection type.

The following IBM Data Cataloging containers require updates for Scale scanning through secondary VLAN as follows:

- `isd-connmgr-main`
- `isd-consumer-scale-le`
- `isd-consumer-scale-scan`

Assign a unique IP address for each container pod set.

For example:

Container	IP address	Resource
`isd-connmgr-main`	10.100.100.11	`Statefulset`
`isd-consumer-scale-le`	10.100.100.12	`Deployment`
`isd-consumer-scale-scan`	10.100.100.13	`Deployment`

```
apiVersion: apps/v1
metadata:
  name: isd-connmgr-main
....
spec:
  template:
    annotations:
        k8s.v1.cni.cncf.io/networks: |-
          [
            {
              "name": "br4001-c",
              "interface": "br4001",
              "namespace": "ibm-data-cataloging",
              "ips": ["10.100.100.11/24"]
            }
          ]
```

Activate changes

1. Do a graceful shutdown of the IBM Data Cataloging service. For the procedure, see Graceful shutdown.
2. Edit `Statefulset isd-conn-mgr-main` in the components. You can append the new annotation to the current annotations.
3. Edit `Deployments` for the consumer type. In this case it depends on the connection type.
   For example for scale it must be the `isd-consumer-scale-scan`.
4. Return the IBM Data Cataloging service to the running state. For the procedure, see Returning IBM Data Cataloging to a running state.

Example Scaling Commands:

- Scale down IBM Data Cataloging:

```
oc get deployments |grep consumer
oc get deployments |grep consumer|awk '{ print $1}'|xargs -L 1 echo oc scale --\ replicas=0 deployment |bash

oc get deployments |grep producer
oc get deployments |grep producer|awk '{ print $1}'|xargs -L 1 echo oc scale --\ replicas=0 deployment |bash

oc get deployments |grep isd-producer-scale-le
oc get deployments |grep isd-producer-scale-le|awk '{ print $1}'|xargs -L 1 echo oc \ scale --replicas=0 deployment
|bash

oc get statefulset |grep connmgr
oc scale --replicas=0 statefulset.apps/isd-connmgr-main

oc get deployments |grep db2whrest
oc scale --replicas=0 deployment db2whrest
```

- Scale up IBM Data Cataloging:

```
$oc scale --replicas=1 deployment <deployment_name isd-db2whrest
$oc get deployments |grep db2whrest
isd-db2whrest                    1/1     1            1           6d20h

oc get deployments |grep consumer
oc get deployments |grep consumer|awk '{ print $1}'|xargs -L 1 echo oc scale --\ replicas=10 deployment |bash

oc get deployments |grep producer
oc get deployments |grep producer|awk '{ print $1}'|xargs -L 1 echo oc scale --\ replicas=10 deployment |bash

oc get deployments |grep connmgr
oc scale --replicas=1 statefulset.apps/isd-connmgr-main
```

VLAN Testing

Use the following commands to verify that the VLAN is present in the containers.

Note: Addresses are hex byte reverse.

```
sh-4.4# cat /proc/net/route
Iface   Destination     Gateway   Flags   RefCnt  Use   Metric  Mask   MTU     Window  IRTT

br4001 0064640A         00000000  0001      0       0      0    00FFFFFF  0       0       0
br4001 0064640A         0164640A  0003      0       0      0    00FFFFFF  0       0       0
sh-4.4#
```

This is what expecting to see for the `10.100.100.0` network through `10.100.100.1` router.

A utility container can be used with more commands like **apline** or **centos** to confirm that the network endpoints are reachable.

Using **/proc** to determine assigned IP address.

```
sh-4.4# cat /proc/net/fib_trie | grep "|--" | egrep -v "0.0.0.0"
                    |-- 10.100.100.0
                    |-- 10.100.100.11
...
             |-- 127.0.0.0
            |-- 127.0.0.1
                    |-- 10.100.00.0
                    |-- 10.100.100.11
...
             |-- 127.0.0.0
            |-- 127.0.0.1
         |-- 127.255.255.255
```

Testing pods:

From the virtual machine or target cluster, ping the container or pod IP through VLAN.

After completion of all steps, create IBM Data Cataloging data connection. For the procedure, see Creating an IBM Storage Scale data source connection.

Multiple connection managers

Multiple connection managers are a new capability that is designed to enhance scanning performance and enable parallel ingestion. It proves especially valuable in scenarios where data sources are geographically dispersed and need to be scanned as remote sources.

With this new capability, users can take advantage of two primary deployment scenarios:

- The first scenario involves deploying multiple connection managers within the same cluster, allowing for efficient coordination and distribution of scanning tasks. It enables optimized resource utilization and faster processing of data.
- The second scenario involves adding external nodes to the system and deploying one or more connection managers on these nodes. This distributed setup further enhances scanning performance by using extra computing resources and enabling parallel execution of scanning operations across multiple nodes.
  Note:
  - Multiple connection managers improve scanning performance, but it is necessary to understand that the increase in scanning speed does not increase in the performance of indexing records into the database.
    The indexing process might have its own limitations and dependencies that might impact overall performance.

  Example deployment scenario:
  - Main cluster located in Mexico with 2 connection managers deployments.
  - One remote worker located in France with 3 connection manager deployed to scan France data sources.
  - One remote worker located in Canada with 2 connection manager deployed to scan Canada data sources.
    Example:

    ```
    kind: SpectrumDiscover
    apiVersion: spectrum-discover.ibm.com/v1alpha1
    metadata:
      name: spectrumdiscover-sample
      namespace: discover
    spec:
      license:
        accept: true
      doInstall: true
      rwx_storage_class: ibmc-file-gold-gid
      connmgr:
        site: mexico
        replicas: 2
        extraLocations:
          - site: france
            locationType: remote
            replicas: 2
          - site: canada
            locationType: remote
            replicas: 3
            affinity:
      tolerations:
      - effect: PreferNoSchedule
        key: isd
        operator: Exists
    ```

    In case it's required to modify main location needs to be specified it onsite property as follows:

    ```
    connmgr:
        site: france
    ```

  Note: In case site does not exist in a statefulset as example.com or empty, for instance, internal scheduler assigns it to any type local connection manager available.

Enabling feature for skipping the snapshot directories

When the skipping feature is enabled, you can skip the metadata ingestion to the snapshot directories.

Before you begin

- You must set the following environment variables in the IBM Data Cataloging:
  - *NFS_SKIP_SNAPSHOT_DIRS*
  - *INCLUDE_SCALE_SNAPSHOTS*
- Set the environment variable by using the configmap '`connmgr`'.

About this task

To enable the feature for skipping *NFS_SKIP_SNAPSHOT_DIRS* the snapshot directories, you need to set the following environment variable in IBM Data Cataloging.

NFS_SKIP_SNAPSHOT_DIRS
: This environment variable is used specifically for an NFS connection. When the variable is set to 'True', IBM Data Cataloging skips the scanning of snapshot directories. When the variable is set to 'False' or not set at all, it scans the snapshot directories.

INCLUDE_SCALE_SNAPSHOTS
: When the *INCLUDE_SCALE_SNAPSHOTS* variable value is set to 'false' (default value), the IBM Data Cataloging scan excludes all the files that are inside the .snapshots directories, otherwise, if the variable value is set to 'true', the scan includes all the files, including the .snapshots directories.

- **Enabling skip snapshot directories feature on Red Hat OpenShift**
  The following procedure helps you set the configuration for skipping snapshot directories by using *NFS_SKIP_SNAPSHOT_DIRS* and *INCLUDE_SCALE_SNAPSHOTS* variable.

Enabling skip snapshot directories feature on Red Hat® OpenShift®

The following procedure helps you set the configuration for skipping snapshot directories by using *NFS_SKIP_SNAPSHOT_DIRS* and *INCLUDE_SCALE_SNAPSHOTS* variable.

Before you begin

Add the following variables as needed to the configmap and set the value to 'True'.

- *NFS_SKIP_SNAPSHOT_DIRS*
- *INCLUDE_SCALE_SNAPSHOTS*

Procedure

1. Issue the following command to change the current project:

   ```
   oc project ${PROJECT_NAME}
   ```

2. Issue the following command to edit the configmap:

   ```
   oc edit configmap connmgr
   ```

   After you run the preceding command, the output looks similar to the following as shown here. You can modify the following file and add or update the variables and its value in the `data` section.

   ```
   # Please edit the object below. Lines beginning with a '#' will be ignored,
   # and an empty file will abort the edit. If an error occurs while saving this file will be
   # reopened with the relevant failures.
   #
   apiVersion: v1
   data:
     CONNMGR_ENDPOINT: /connmgr/v1/
     CONNMGR_PROTOCOL: http
     NFS_SKIP_SNAPSHOT_DIRS: "True"
   kind: ConfigMap
   metadata:
     creationTimestamp: "2022-08-26T19:55:39Z"
     name: connmgr
     namespace: spectrum-discover
     ownerReferences:
     - apiVersion: spectrum-discover.ibm.com/v1alpha1
       kind: SpectrumDiscover
       name: spectrumdiscover
       uid: 001fb331-c2ef-4b28-8827-5c7d6a702904
     resourceVersion: "13761646"
     uid: bf621847-4c00-44c9-a7f2-4d3430ede67b
   ```

3. Save the preceding updated object and issue the following command to verify that data is updated.

   ```
   oc get configmap connmgr -o yaml
   ```

   After you run the preceding command, the output looks similar to the following as shown here:

```
apiVersion: v1
data:
  CONNMGR_ENDPOINT: /connmgr/v1/
  CONNMGR_PROTOCOL: http
  NFS_SKIP_SNAPSHOT_DIRS: "True"
kind: ConfigMap
metadata:
  creationTimestamp: "2022-08-23T11:33:40Z"
  name: connmgr
  namespace: spectrum-discover
  ownerReferences:
  - apiVersion: spectrum-discover.ibm.com/v1alpha1
    kind: SpectrumDiscover
    name: spectrumdiscover
    uid: 64aa30c1-d621-47f5-9ee8-763ee9386bbd
  resourceVersion: "293269249"
  uid: e54ebffa-be11-4d05-82e4-03bc94edbce6
```

4. Issue the following command to verify that the update is made in the connection manager pod:

```
oc -n ibm-data-cataloging delete sts/isd-connmgr-main
```

Managing IBM Data Cataloging rules when integrated with CAS

IBM Data Cataloging provides the ability to integrate with Content-Aware Storage to configure file ingestion filtering. This feature enables you to filter files from a data source based on specific criteria, such as file type or content, and ingest them into the corresponding domain.

1. Ensure that you have IBM Fusion 2.11.0 version.
2. Install the IBM Data Cataloging to 2.3.0 version. For more information, see Installing IBM Data Cataloging.
   Important: If you face any issues during the installation of the IBM Data Cataloging 2.3.0, then see Data Cataloging upgrade issues to resolve the issue.
3. Connection to Scale Remote Filesystem from IBM Data Cataloging with live events enabled. For more information, see IBM Storage Scale data source connections.
4. Configure file-ingestion filtering using IBM Data Cataloging on the Content-Aware Storage. For more information, see Integrating IBM Data Cataloging for file filtering.
5. When IBM Fusion Data Cataloging is enabled to perform file filtering on Content-Aware Storage, a default rule is automatically created. This rule filters files based on the file types such as **pdf**, **doc**, **docx**, **txt**, **html**, **markdown**, **md**, **pptx**, **jpeg**, **png**, **bmp**, **xslx**, **asciidoc**, **adoc**, **xhtml**, **csv**, **webp**, **wav**, **ogg**, **opus**, **mp3**.
   For example:

```
apiVersion: datacataloging.ibm.com/v1alpha1
metadata:
  name: rule-allowed-filetypes
  namespace: ibm-data-cataloging
spec:
  metadata:
    - field: filetype
      operator: in
      value:
        - pdf
        - doc
        - docx
        - txt
        - html
        - markdown
        - md
        - pptx
        - jpeg
        - png
        - bmp
        - xslx
        - asciidoc
        - adoc
        - xhtml
        - csv
        - webp
        - wav
        - ogg
        - opus
        - mp3
```

6. Use the following command to extend the list of supported files, you can update the default rule by patching the current Rule object.
   For example, to include the file types such as **pdf**, **txt**, **html**, **json**, **md**, **docx**, **pptx**, **bmp**, **jpeg**, **png**, **tiff**, and **sh**.

```
DCS_NS=ibm-data-cataloging
oc patch rule allowed-filetypes -n $DCS_NS --type merge -p '{"spec": {"metadata": [{"field": "filetype","operator":
"in","value": ["doc", "docx", "pdf", "txt", "gif", "html", "json", "md", "pptx", "bmp", "jpeg", "png", "tiff", "sh" ]}]}}'
```

7. In addition to metadata rule filtering, you can perform a complementary filter based on content using regex-based content search capability. For more information, see Identifying the required regex expressions.
   For example, to filter files based on the presence of an email pattern, you can add a content requirement to the default rule:

```
DCS_NS=ibm-data-cataloging
oc -n $DCS_NS patch rule allowed-filetypes --type=merge -p '{"spec":{"content":>
[{"field":"email","operator":"regex","value":"\b[\w\.=-]+@[\w\.-]+\.[\w]{2,3}\b"}]}}'
```

Administering IBM Data Cataloging

This section provides information on how to administering the IBM Data Cataloging capabilities.

- **AI-enabled capabilities**
  IBM Fusion Data Cataloging is expanding its AI-driven data management capabilities with two significant innovations. The new IBM Data Cataloging Model Context Protocol (MCP) server that enables multiple large language models (LLMs) to connect to and operate across a unified set of cataloging tools.
- **Managing user access**
  The Data cataloging environment provides access to users and groups. The role that is assigned to a user or group determines the functions that are available. Users and groups can also be associated with collections that use policies that determine the metadata that is available to view.
- **Managing metadata policies**
  Policies might be used to automatically tag a set of documents on a periodic basis. In addition, policies might be used to send sets of documents to be deep-inspected by a registered application.
- **Using content search policies**
  You can define regular expressions to search for and create policies that use the regular expressions.
- **Tiering data by using ScaleILM application**
  Use the IBM Data Cataloging ScaleILM application to move data to different tiers (pools) that are configured on the IBM Storage Scale connection.
- **Copying data by using ScaleAFM application**
  The ScaleAFM application supports copy function between IBM Storage Scale connection and IBM Cloud® Object Storage connection.
- **Exporting metadata to IBM Watson Knowledge Catalog**
  IBM Data Cataloging Watson Knowledge Catalog (WKC) connector supports export of the metadata to either an IBM Cloud instance or to an On-Premise Instance of the Watson Knowledge Catalog .
- **Importing externally curated tags for COS/S3 using import tags application**
  The import tags application is used to import a set of externally curated tags for Cloud Object Storage and S3 services.
- **Performing retention analytics on IBM Storage Protect archive data**
  Use IBM Data Cataloging to perform retention analytics on archive data managed by IBM Storage Protect.
- **Managing tags**
  A tag is a custom metadata field used to enhance system metadata with organization-specific information. For instance, an organization may categorize its storage by project, cost center, or region but the attributes not provided by the storage system itself. Tags enable data classification, search, and reporting based on these organizational dimensions. Use tags to store this additional information and to manage or locate data using metadata that reflects your organizational context.
- **Discover data**
  By discovering your data, you can apply policies that assign tags to your data. You can apply tags to the results of a single search, or you can use policies to automatically apply tags on a periodic basis.
- **Managing applications**
  An application is a program that interfaces with IBM Data Cataloging and can access the source storage. There are many use cases for application, including data content inspection for enriching metadata, data movement or migration, data scrubbing or sanitation, and more. Data is identified by IBM Data Cataloging by policy filter and passed to the application as pointers through a messaging queue. Then, the application performs whatever work is appropriate on the source data and returns a completion status back to IBM Data Cataloging, which might or might not include enriched metadata for the records. If it does include enriched metadata, IBM Data Cataloging catalogs that metadata and makes it immediately searchable.
- **Reports**
  Reports can be generated upon applying tags to a set of data.
- **Creating a IBM Data Cataloging application for metadata-based policies**
  Provides information about how to create IBM Data Cataloging application for metadata-based policies.
- **Graceful shutdown**
  Provides detailed instructions to put IBM Data Cataloging in idle state on an OpenShift® environment.
- **REST API for IBM Data Cataloging**
  REST API reference documentation provides endpoint specifications, request and response formats, and authentication details for programmatic access to IBM Spectrum Discover services.
- **Data protection**
  Data protection features in IBM Spectrum Discover support backup, disaster recovery, and restore operations to protect catalog metadata and system configurations.
- **Using a third-party data movement application to manage data**
  Introduction to data movement with IBM Data Cataloging.

AI-enabled capabilities

IBM Fusion Data Cataloging is expanding its AI-driven data management capabilities with two significant innovations. The new IBM Data Cataloging Model Context Protocol (MCP) server that enables multiple large language models (LLMs) to connect to and operate across a unified set of cataloging tools.

Complementing this, IBM Fusion Data Cataloging is introducing a technology preview of Content-based AI-enabled autotagging, which enables you to selectively analyze data and automatically generate tags based on document content.

Together, these capabilities enhance governance, automation, and intelligent discovery across enterprise data ecosystems.

- **IBM Data Cataloging MCP server**
  IBM Data Cataloging Service offers a new capability of a Model Context Protocol (MCP) server. It is designed to enable seamless interoperability between multiple large language models (LLMs) and a unified suite of cataloging and data management tools. Through a centralized control layer, the MCP server allows LLMs to securely connect, interact, and run operations by using tools that are registered within the MCP environment.
- **Content-based AI-enabled autotagging**
  IBM Data Cataloging provides AI-powered content-based tagging to automatically classify unstructured data by using the actual content of files or objects, not only their metadata. This capability uses an inference model that is accelerated by external GPUs to propose and apply business-relevant tags across large datasets with minimal manual effort.

IBM Data Cataloging MCP server

IBM Data Cataloging Service offers a new capability of a Model Context Protocol (MCP) server. It is designed to enable seamless interoperability between multiple large language models (LLMs) and a unified suite of cataloging and data management tools. Through a centralized control layer, the MCP server allows LLMs to securely connect,

interact, and run operations by using tools that are registered within the MCP environment.

Integrated with IBM Data Cataloging, the MCP server enhances how AI-driven systems access, analyze, and organize metadata from diverse data sources. It provides a consistent interface for tool discovery, execution, and lifecycle management, reducing the need for direct integrations between individual LLMs and the underlying cataloging infrastructure.

Key capabilities

Natural language queries on Data Cataloging Services (DCS)
: It enables the user to explore and query cataloged information by using natural language, without requiring knowledge of query syntax. This capability improves data accessibility and empowers business users to retrieve insights more intuitively.

LLM-based tags and labels catalog suggestions
: It assists the user in creating, managing, and discovering tags more efficiently. The LLM can use the MCP tools to suggest relevant tags based on dataset content and highlight existing tags across the catalog. This capability helps maintain metadata consistency, reduces manual effort, and enhances the overall accuracy of data classification.

Policy-driven auto-tagging and management
: It enables you to view existing catalog policies and create or run auto-tagging policies that are powered by AI. These policies automatically classify and label datasets based on predefined rules and model insights. This helps enforce governance standards, improves compliance, and ensures that metadata remains accurate and up to date over time.

Before you begin

Before you enable and use the MCP server within the IBM Data Cataloging environment, ensure that your system meets the following requirements. These prerequisites are necessary to authenticate with your cluster, enable the MCP components, and verify the available service endpoints.

Ensure that the following tools are installed and accessible from your local environment:

- Red Hat OpenShift® CLI (`oc`)
- `curl`

After verifying the prerequisites, log in to your OpenShift cluster by using CLI.

Setting up the MCP server

To enable MCP server capabilities, ensure that the Data Catalog has the required features that are enabled for the MCP server.

Note: The `DCS_NS` variable specifies the namespace where the DCS instance is deployed. By default, it is `ibm-data-cataloging`, and adjusts according to the namespace where it is installed.
Run the following command:

```
DCS_NS=ibm-data-cataloging

oc -n ${DCS_NS} patch SpectrumDiscover $(oc -n ${DCS_NS} get spectrumdiscover -o jsonpath='{.items[*].metadata.name}') \
  --type=merge -p '{
    "spec": {
      "enabled_features": {
        "natural-language-queries": {
          "enabled": true
        }
      }
    }
  }'
```

After the patch is successfully applied, the `SpectrumDiscover` loads the MCP server with new capabilities.

Verifying that MCP server is running
: After enabling the features, verify that the MCP server pods are up and running.

  Run the following command:

  ```
  oc -n ${DCS_NS} get pod -l 'app=isd,role=dcs-ai-mcp-server'
  ```

  Example output:

  ```
  isd-dcs-ai-mcp-server 1/1 Running 0 3h
  ```

  If you see pods in the Running state, the MCP server is successfully up and operational.

Connecting to MCP server

After the MCP server is deployed and prerequisites are met, you can retrieve the service endpoints to enable LLMs or interact with its tools. The MCP server exposes both SSE and HTTP endpoints for integration.

Retrieve MCP endpoints on Linux/macOS

```
oc get routes -n ${DCS_NS} -o json \
  | jq -r '.items[]
  | select(.metadata.name | test("mcp"))
  | "Route: \(.metadata.name)\nHost: \(.spec.host)\nSSE Endpoint: https://\(.spec.host)/mcp\nHTTP Endpoint: https://\
(.spec.host)/mcp/http\n"'
```

Example output:

```
Route: dcs-mcp-route
Host: dcs-mcp-route-ibm-data-cataloging.apps.com
SSE Endpoint: https://dcs-mcp-route-ibm-data-cataloging.ibm.com/mcp
HTTP Endpoint: https://dcs-mcp-route-ibm-data-cataloging.com/mcp/http
```

The command outputs -

- `Route`: The OpenShift route for the MCP server.
- `Host`: The host URL assigned by OpenShift.
- `SSE Endpoint`: The URL to connect through server-sent events.
- `HTTP Endpoint`: The URL to connect through standard HTTP requests.
  This connection provides the necessary endpoints to integrate LLMs or other clients with the MCP server.

Initialization

Example of how to initialize the MCP server:

```
curl -i -X POST \
  <MCP http endpoint> \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{
    "jsonrpc": "2.0",
    "id": 1,
    "method": "initialize",
    "params": {}
  }' --insecure
```

Replace all placeholder values before sending the request.

Example Output:

```
HTTP/1.1 200 OK
access-control-allow-headers: Content-Type
access-control-allow-methods: GET, POST, OPTIONS
access-control-allow-origin: *
cache-control: no-cache, no-transform
content-type: text/event-stream
mcp-session-id: xxxxxxxxxxxxxxxxxxx
date: Tue, 16 Dec 2025 19:17:50 GMT
transfer-encoding: chunked
set-cookie: xxxxxxxxxxxxxxxxxxxxxxx; path=/; HttpOnly; Secure; SameSite=None
```

Save the `mcp-session-id`, it is necessary to send requests.

List Tools

```
curl -X POST \
  <MCP http endpoint> \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -H "mcp-session-id: <mcp-session-id>" \
  -d '{
    "jsonrpc": "2.0",
    "id": 1,
    "method": "tools/list",
    "params": {}
  }' \
  --insecure
```

Replace all placeholder values before sending the request.

MCP server - Tools

The following section describes the new tools that are introduced in the MCP server, which is designed to improve interaction with data sources, manage credentials, policies, and tagging. Each tool serves a specific purpose and can be used independently or together to optimize data management and automation workflows.

`dcs_set_credentials`

The Set Credentials tool enables you to configure the credentials that are required to connect the MCP server to the Data Cataloging Service.

Tool name	Arguments	Request
`dcs_set_credentials`	DCS URL, Username, Password	`curl -X POST <mcp http endpoint> -H "Content-Type: application/json" -H "Accept: application/json, text/event-stream" -H "mcp-session-id: <mcp-session-id>" -d '{ "jsonrpc": "2.0", "id": 1, "method": "tools/call", "params": { "name": "dcs_set_credentials", "arguments": { "cluster_url": "<DCS URL>", "username": "user", "password": "password" } } }' --insecure`
	DCS URL, Authentication token	`curl -X POST <mcp http endpoint> -H "Content-Type: application/json" -H "Accept: application/json, text/event-stream" -H "mcp-session-id: <mcp-session-id>" -d '{ "jsonrpc": "2.0", "id": 1, "method": "tools/call", "params": { "name": "dcs_set_credentials", "arguments": { "cluster_url": "<DCS URL>", "token": "<token>" } } }' --insecure`

Usage: It is used when starting a new session or when you need to update your connection credentials.

Replace all placeholder values before sending the request.

Obtaining a username and password:

You can retrieve the user and password by using the following commands:

macOS

```
#User
oc get secret keystone -n ${DCS_NS} -o jsonpath="{.data.user}" | base64 -D; echo

#Password
oc get secret keystone -n ${DCS_NS} -o jsonpath="{.data.password}" | base64 -D; echo
```

Linux

```
#User
oc get secret keystone -n ${DCS_NS} -o jsonpath="{.data.user}" | base64 --decode; echo

#Password
oc get secret keystone -n ${DCS_NS} -o jsonpath="{.data.password}" | base64 --decode; echo
```

Obtaining DCS URL:
Retrieve the DCS URL by using the following command:

```
DCS_CONSOLE=$(oc -n ${DCS_NS} get route console -o jsonpath='{"https://"}{.spec.host}')
echo $DCS_CONSOLE
```

Obtaining token:
Retrieve the token by using the following command:

```
curl -k -I -u <username>:<password> ${DCS_CONSOLE}/auth/v1/token | grep x-auth-token
```

Response:

```
x-auth-token: <token>
```

`dcs_file_search`

The File Search tool enables you to search across cataloged data sources by using natural language queries. It automatically converts user input into structured queries that retrieve relevant files, datasets, or metadata entries.

With this capability, you can find information quickly without needing to understand query syntax or underlying database languages. By bridging natural language with structured search, File Search improves accessibility for both technical and non-technical users, and accelerates data discovery.

Tool name	Argument	Request

`dcs_file_search`	Query	`curl -X POST`
		`<mcp http endpoint>`
		`-H "Content-Type: application/json"`
		`-H "Accept: application/json, text/event-stream"`
		`-H "mcp-session-id: <mcp-session-id>"`
		`-d '{ "jsonrpc": "2.0", "id": 1, "method": "tools/call", "params": { "name":`
		`"dcs_file_search", "arguments": { "query": "SELECT * FROM METAOCEAN LIMIT 1" } } }'`
		`--insecure`

Replace all placeholder values before sending the request. Replace the query as needed.

Example prompt:

```
Show all CSV files containing customer transactions from 2023
```

The tool retrieves matching entries from the registered data sources.

`dcs_get_registered_tags`

The Get Registered Tags tool retrieves a complete list of all tags that are registered in the cataloging system. It helps users and AI agents review existing classifications before creating or suggesting new ones, ensuring that tag definitions remain consistent and non-duplicative.

This tool is useful for maintaining a clean and standardized tagging structure across multiple datasets and departments.

Tool name	Request

`dcs_get_registered_tags`	`curl -X POST`
	`<mcp http endpoint>`
	`-H "Content-Type: application/json"`
	`-H "Accept: application/json, text/event-stream"`
	`-H "mcp-session-id: <mcp-session-id>"`
	`-d '{ "jsonrpc": "2.0", "id": 1, "method": "tools/call", "params": { "name":`
	`"dcs_get_registered_tags", "arguments": { } } }'`
	`--insecure`

Replace all placeholder values before sending the request.

Example use case:

Before adding a new Financial tag, you can check whether a similar tag exists in the catalog.

`dcs_get_recommend_tags`

The Recommend Tags tool uses LLM-powered analysis to suggest relevant tags for cataloged files and datasets. By examining metadata and contextual information, the tool recommends tags that help classify data more accurately and consistently.

This feature significantly reduces the time that is spent on manual tagging and enhances catalog quality by ensuring that all datasets are properly labeled.

Tool name	Argument	Request

Tool name	Argument	Request
`dcs_get_recommend_tags`	Query	`curl -X POST <mcp http endpoint> -H "Content-Type: application/json" -H "Accept: application/json, text/event-stream" -H "mcp-session-id: <mcp-session-id>" -d '{ "jsonrpc": "2.0", "id": 1, "method": "tools/call", "params": { "name": "dcs_get_recommend_tags", "arguments": { "query": "SELECT * FROM METAOCEAN LIMIT 10" } } }' --insecure`

Replace all placeholder values before sending the request. Replace the query as needed.

Example prompt:

```
Recommend me some tags to classify my CSV files
```

The tool retrieves tags recommendation based on the metadata.

`dcs_create_tag`

The Create Tag tool enables you to define and create new tags.

Tool name	Arguments	Request
`dcs_create_tag`	`tagName`, `type`	`curl -X POST <mcp http endpoint> -H "Content-Type: application/json" -H "Accept: application/json, text/event-stream" -H "mcp-session-id: <mcp-session-id>" -d '{ "jsonrpc": "2.0", "id": 1, "method": "tools/call", "params": { "name": "dcs_create_tag", "arguments": { "tagName": "TEST", "type": "VARCHAR(256)" } } }' --insecure`

Replace all placeholder values before sending the request.

Example prompt:

```
Create the tags that you recommend me.
```

The tool creates tags based on the recommendations.

`dcs_get_policies`

The Get Policies tool retrieves and lists all existing cataloging policies within the system. Policies define automated rules or governance conditions that are applied to data, metadata, or tags.

This tool provides visibility into the active policies that govern cataloged assets. It also helps users understand which rules are being enforced and how they affect data classification or compliance workflows.

Tool name	Request
`dcs_get_policies`	`curl -X POST <mcp http endpoint> -H "Content-Type: application/json" -H "Accept: application/json, text/event-stream" -H "mcp-session-id: <mcp-session-id>" -d '{ "jsonrpc": "2.0", "id": 1, "method": "tools/call", "params": { "name": "dcs_get_policies", "arguments": { } } }' --insecure`

Replace all placeholder values before sending the request.

Example use case:

A data steward can use this tool to view all current auto-tagging and compliance policies before creating new ones.

`dcs_create_policy`

The Create AutoTag Policy tool enables you to define and deploy new auto-tagging policies that automatically assign tags to cataloged assets based on predefined rules and AI insights. It streamlines governance by automating repetitive classification tasks and ensuring that policies remain consistent across data sources.

This tool currently supports auto-tag policy types, allowing AI models to dynamically apply tags based on metadata patterns or user-defined contextual rules.

Tool name	Arguments	Request
`dcs_create_policy`	`polFilter`, `policyName`, `schedule`, `tags`	`curl -X POST <mcp http endpoint> -H "Content-Type: application/json" -H "Accept: application/json, text/event-stream" -H "mcp-session-id: <mcp-session-id>" -d '{ "jsonrpc": "2.0", "id": 1, "method": "tools/call", "params": { "name": "dcs_create_policy", "arguments": { "polFilter": "filetype='pdf'", "policyName": "TEST", "schedule": "NOW", "tags": {'tagtest':'valtest'} } }' --insecure`

Replace all placeholder values before sending the request.

Example prompt:

```
Explore the catalog for security-related flaws and recommend me 5 different tags with proposed values that can be used for identify and classify the data on the catalog, make sure they are not duplicate.
```

Content-based AI-enabled autotagging

IBM Data Cataloging provides AI-powered content-based tagging to automatically classify unstructured data by using the actual content of files or objects, not only their metadata. This capability uses an inference model that is accelerated by external GPUs to propose and apply business-relevant tags across large datasets with minimal manual effort.

Overview

Traditional metadata-based tagging relies on file attributes such as name, size, or type. AI-powered content-based tagging goes further: the system reads file content, combines it with a defined business context and then proposes the most relevant tag names and values by consuming a reasoning model.

The feature operates in two stages:

Stage 1: Tag proposal (dry run)
: The system analyzes a sample of files in a pre-configured path, generates tag name suggestions that are aligned with the defined business context, and stores them in the business context ConfigMap. No tags are created or applied in IBM Data Cataloging service at this point. Review the proposal and activate the ones that are approved.

Stage 2: Tag application
: On subsequent runs, the system detects the tags that are activated and proceeds to apply them: It creates the tags in IBM Data Cataloging if they do not exist already, retrieves untagged files or objects from the catalog, uses the reasoning model to determine the appropriate value for each active tag, and enriches the catalog.

The two stages are controlled entirely through the business context ConfigMap and run automatically on a 30-minute schedule. This approach enables consistent tags assignments with no manual categorization, standarizing data management across large data sources and improves discovery and governance capabilities.

Before you begin

Before enabling this feature, ensure that the following prerequisites are met:

- IBM Fusion 2.13 is deployed with access to the IBM Storage Scale filesystem through a Global Data Platform remote mount.
- IBM Data Cataloging is installed at 2.5.3 version and a IBM Data Cataloging service connection to an IBM Storage Scale filesystem is configured and online.
- An AI inference engine is accessible from the cluster.
- The `ai-suggestions` feature toggle is available in your `SpectrumDiscover` custom resource.

Activating content-based autotagging

To activate content-based autotagging, follow these steps:

1. Create the business context ConfigMap.
   Create a ConfigMap named `ai-tag-extraction-context` in the IBM Data Cataloging namespace. This ConfigMap defines the data source, the path to analyze, and the business context that guides the reasoning model.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: ai-tag-extraction-context
  namespace: ibm-data-cataloging
data:
  connection_names: "<dcs-connection-name>"
  filesystem: "<storage-scale-filesystem-cr-name>"
  path: "/gpfs/<filesystem>/path/to/your/dataset"
  allowed_filetypes: "pdf,txt,docx"
  context: "<<Describe your business use case here. The more specific the context,
    the more relevant the proposed tags. For example: This dataset contains
    HR profiles including job role, employment status, and salary information.
    The goal is to classify these files for talent management and compliance.>>"
```

Field	Description
`connection_names`	The name of the IBM Data Cataloging service connection as registered in the catalog.
`filesystem`	Full name of the `filesystems.scale.spectrum.ibm.com` custom resource in the `ibm-spectrum-scale` namespace. Run `oc get filesystems.scale.spectrum.ibm.com -n ibm-spectrum-scale` to find the correct name.
`path`	Full IBM Storage Scale path to the directory containing the files to tag.
`allowed_filetypes`	Comma-separated list of file extensions to process.
`context`	Business context description that guides the reasoning model tag proposals.

2. Enable the feature in the `SpectrumDiscover` custom resource.
   Edit the `SpectrumDiscover` custom resource to enable both the reasoning model proxy and the AI suggestions feature:

```
enabledFeatures:
  aiSuggestions:
    enabled: true
    inferenceEngineHost: "http://<reasoning-model-host>:<port>"
    model: "granite4"
```

Field	Description
`inferenceEngineHost`	URL of the reasoning or compatible inference engine accessible from the cluster.
`model`	Model alias to use. The default is `granite4`.

   After saving the custom resource, the operator deploys the following resources automatically:
   - LLM proxy deployment and service (`isd-llm-proxy`)
   - Remote content search deployment and service (`isd-rcs-ai-tags`)
   - AI tag extraction CronJob (`isd-ai-tag-extraction`, runs every 30 minutes)

Using content-based autotagging

First run: tag proposal
: On the first CronJob execution, the system performs a dry run:

1. Reads the business context ConfigMap.
2. Analyzes a sample of files in the configured path by using the reasoning model.
3. Proposes tag names and their possible values.
4. Writes the proposals to a new key `suggested_tags` in the same ConfigMap.
5. Exits without creating or applying any tags in the IBM Data Cataloging service.

After the first run, inspect the ConfigMap to review the proposals:

```
oc get cm ai-tag-extraction-context -n ibm-data-cataloging -o jsonpath='{.data.suggested_tags}' | python3 -m json.tool
```

Example output:

```
{
  "EMPLOYEE_STATUS": { "active": false },
  "JOB_ROLE":        { "active": false },
  "LOCATION":        { "active": false },
  "SALARY_INDICATOR": { "active": false },
  "WORK_ARRANGEMENT": { "active": false }
}
```

On subsequent dry runs without any active tags, the system forwards the existing proposals to the reasoning model. Hence, it does not re-propose the same names. New complementary tags might be added to the list.

Activating tags

Edit the `suggested_tags` value in the ConfigMap to set `"active": true` on the tags you want to apply:

```
oc edit cm ai-tag-extraction-context -n ibm-data-cataloging
```

Change the desired entries:

```
{
  "EMPLOYEE_STATUS": { "active": true },
  "JOB_ROLE":        { "active": true },
  "LOCATION":        { "active": false },
  "SALARY_INDICATOR": { "active": true },
  "WORK_ARRANGEMENT": { "active": false }
}
```

On the next CronJob run (within 30 minutes, or triggered manually), the system:

1. Detects the tags marked `active: true`.
2. Creates those tags in IBM Data Cataloging service if they do not exist.
3. Scans the configured connection to refresh the catalog.
4. Queries for untagged files matching the allowed filetypes and path.
5. Calls the reasoning model to extract tag values.
6. Applies the values to the catalog.

Triggering a manual run

To run the job immediately without waiting for the next scheduled execution:

```
oc create job --from=cronjob/isd-ai-tag-extraction manual-run-$(date +%s) -n ibm-data-cataloging
```

Monitor the run:

```
oc logs -n ibm-data-cataloging -l job-name=manual-run-<timestamp> --follow
```

Monitoring

Verify deployed resources

```
# Business context ConfigMap
oc get cm ai-tag-extraction-context -n ibm-data-cataloging

# Remote content search service (must exist before the CronJob can reach the LLM)
oc get svc isd-rcs-ai-tags -n ibm-data-cataloging

# CronJob schedule and last execution
oc get cronjob isd-ai-tag-extraction -n ibm-data-cataloging

# Recent job runs
oc get jobs -n ibm-data-cataloging -l role=ai-tag-extraction

# RemoteContentSearch CR status
oc describe remotecontentsearch ai-tags -n ibm-data-cataloging
```

Check feature conditions

```
oc get spectrumdiscover -n ibm-data-cataloging \
  -o jsonpath='{.items[0].status.conditions}' | python3 -m json.tool
```

Verify by using `AISuggestions` with `status: "True"` to confirm that the feature is running correctly.

Disabling content-based autotagging

Set `enabled: false` in the `SpectrumDiscover` custom resource:

```
spec:
  enabledFeatures:
```

```
    aiSuggestions:
      enabled: false
```

Troubleshooting

Symptom	Likely cause	Resolution
The `isd-rcs-ai-tags` service does not exist after enabling the feature.	The `filesystem` field in the ConfigMap does not match the exact CR name in `ibm-spectrum-scale` namespace.	1. Run the `oc get filesystems.scale.spectrum.ibm.com -n ibm-spectrum-scale` command. 2. Update the ConfigMap with the exact name.
PVC `isd-rcs-ai-tags` remains in `Pending` state.	A previous PV with the same name is in `Released` state.	Run the `oc delete pv isd-rcs-ai-tags` command and trigger a reconcile by annotating the `ai-tags` `RemoteContentSearch` CR: `oc annotate remotecontentsearch ai-tags -n ibm-data-cataloging reconcile-trigger="$(date +%s)" --overwrite`
CronJob exits with `no valid connections`.	The `connection_names` value does not match an online IBM Storage Scale connection.	Verify that the connection is online and the name matches exactly.
All files fail with `AI response contained no valid tag values`.	The inference engine is unreachable or returned an unexpected response.	Check LLM proxy logs: `oc logs -n ibm-data-cataloging -l role=llm-proxy`

Use cases

Human Resources document classification
: HR departments store large volumes of sensitive documents including contracts, performance reviews, and compliance records. Content-based tagging automatically classifies these files with tags such as `employee_status`, `job_role`, `salary_indicator`, or `compliance_category`, enabling governance and discovery without manual effort.

Workforce analytics
: Data engineers building analytics pipelines can tag files with dimensions such as `location`, `department`, or `contract_type` to create filtered subsets for downstream processing without modifying the source data.

Managing user access

The Data cataloging environment provides access to users and groups. The role that is assigned to a user or group determines the functions that are available. Users and groups can also be associated with collections that use policies that determine the metadata that is available to view.

User and group access can be authenticated by Data Cataloging, a Lightweight Directory Access Protocol (LDAP) server, the IBM Cloud® Object Storage, or using the Red Hat®OpenShift® credentials. If you use Data Cataloging, LDAP, or IBM COS as authentication methods, then it is required to follow the same authentication process by entering the associated username and password to access the Data Cataloging user interface.

If you use Red HatOpenShift as authentication method, then log in by default through Single-Sign On, unless this option is explicitly disabled. The administrator role can manage the user access functions. For more information, see Initial login.

Note: A local user that is created on the Data cataloging system must use a user name and password to log in. Users from an external LDAP or IBM Cloud® Object Storage domain must include the domain name as a prefix to the user name with a forward slash (/), such as `"<domain>/<user>"`. The domain name is the name that is given to the external authentication domain in Data cataloging.

Roles

Roles determine how users and groups can access records or the Data cataloging environment.

If a user or group is assigned to multiple roles, the least restrictive role is used. For example, if a user is assigned to the Data User role but is also included in a group that is assigned to the Data Admin role, that user has the privileges of the Data Admin role.

The following roles are available:

Admin
: This role can create users, groups, and collections. This role can also manage connections to Lightweight Directory Access Protocol (LDAP) and IBM Cloud Object Storage domains. This role can use the Application Management APIs to install, upgrade, or delete Data cataloging applications that use the Data cataloging API service.

Data Admin
: Users with this role can access all metadata that is collected by Data cataloging and is not restricted by policies or collections. This role can also define tags and policies, including policies that assign a collection value to a set of records.
Note:
The built-in `Collection` tag is a special tag. This tag can be set only by users with the Data Admin role. All other tags can be set by any user with the Data User or Data Admin or Collection Admin role.

: Users with this role can also edit local users and local groups and assign roles and collections to users and groups.

Collection Admin
: The Collection Admin role is a bridge between the Data Admin role and the Data User role. Users with the Collection Admin role can:

- Create, update, and delete the policies for the collections that they administer.
- View, update, and delete policies of data users for the collections they administer. They cannot delete a policy if it has a collection that they do not administer.
- Add users to collections that they administer. These data users can access to a particular collection, which means that they can access to the records marked with that collection value.

- List any type of tag and create or modify `Characteristics` tags. They cannot create, modify, or delete `Open` and `Restricted` tags. These permissions are the same as the ones associated with the Data User role.

Collection User
: Users with this role can access metadata that is collected by IBM Data Cataloging, but metadata access can be restricted by the collections that are assigned to users in this role.

  - Users assigned with the Collection User role can:
    - Run scans of collections the user is assigned.
    - View policies of the collections the user is assigned.
    - List any type of tag.
  - Users assigned with the Collection User role cannot:
    - Create, update, and delete any policies.
    - Create, modify, and delete any tags.

Data User
: Users with this role can access metadata that is collected by Data cataloging, but metadata access can be restricted by the collections that are assigned to users in this role. This role can also define tags and policies, based on the collections to which the role is assigned.

Service User
: This role is assigned to accounts for IBM® service and support personnel.

- **Initial login**
  To log in IBM Data Cataloging graphical user interface, use the following information.
- **Resetting the sdadmin password**
  If the `sdadmin` password changes and you forget the password, you can access the keystone container and run the **reset_sdadmin_details.sh** script to reset the password to the original password.
- **Password policies**
  Data cataloging 2.0.2.1 introduces password policies for the local users who are configured in the default authentication domain.
- **Changing password**
  To change the password, you need to provide the existing and new password.
- **Managing user accounts**
  The administrator can add and manage local user accounts, which are authenticated by IBM Data Cataloging. The administrator can also assign local or LDAP and IBM Cloud Object Storage-managed users to roles and collections.
- **Managing groups**
  The administrator can create and manage local groups that are authenticated by IBM Data Cataloging. The administrator can also assign local or Lightweight Directory Access Protocol (LDAP) and IBM Cloud Object Storage system-managed groups to roles and collections.
- **Managing collections**
  Collections are logical groups of metadata that share a common member access list. For example, a collection can restrict metadata within a research project to the members of the project only. Members outside of the project cannot see the metadata.
- **Managing LDAP and IBM Cloud Object Storage System connections**
  The administrator can create and manage connections to LDAP or IBM Cloud Object Storage System servers that provide authentication for IBM Data Cataloging users.

Initial login

To log in IBM Data Cataloging graphical user interface, use the following information.

IBM Data Cataloging service provides SSO capability. To know more about SSO, see https://www.ibm.com/topics/single-sign-on .

Use your OpenShift® Container Platform credentials to log in IBM Data Cataloging user interface by using SSO.

When Single Sign-On (SSO) is disabled as the default authentication method, the following default login credentials can be used for the IBM Data Cataloging graphical user interface:

Username
: `sdadmin`

Password
: `Passw0rd`

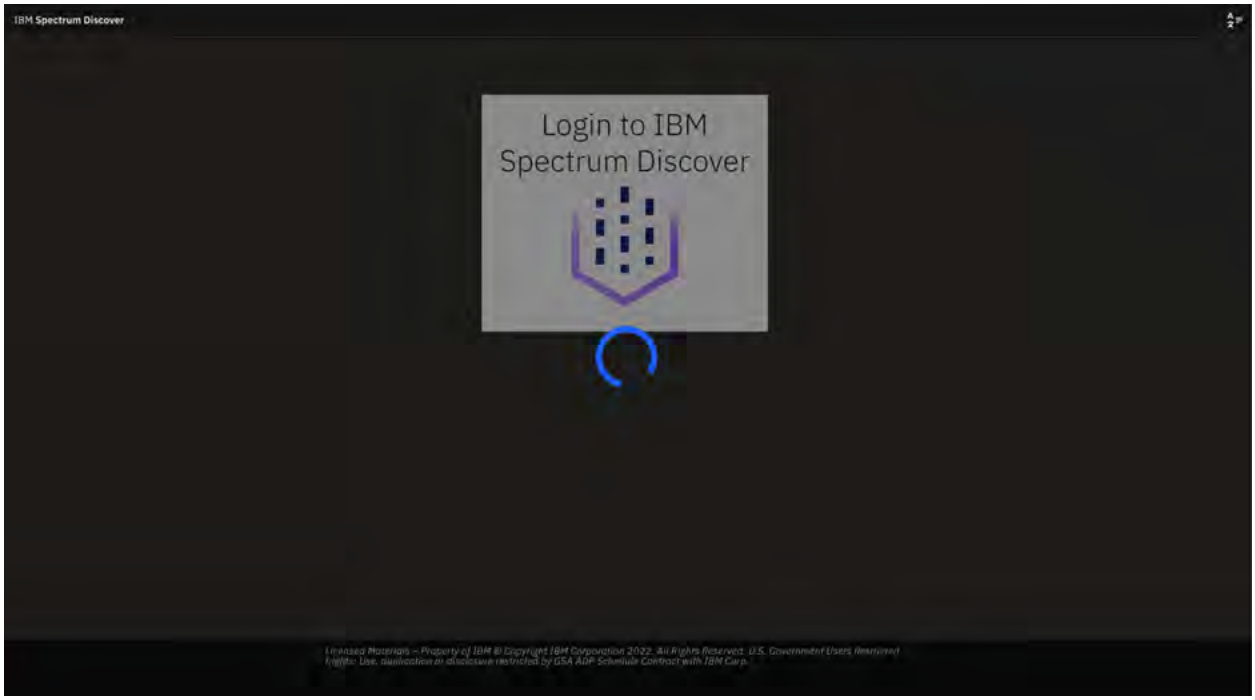
Important:

- When the SSO is disabled, the administrator must change the password during the initial login.
- If you have been using SSO login for a long time, it is not recommended to switch to traditional USERNAME/PASSWORD login because you can face authentication issues.

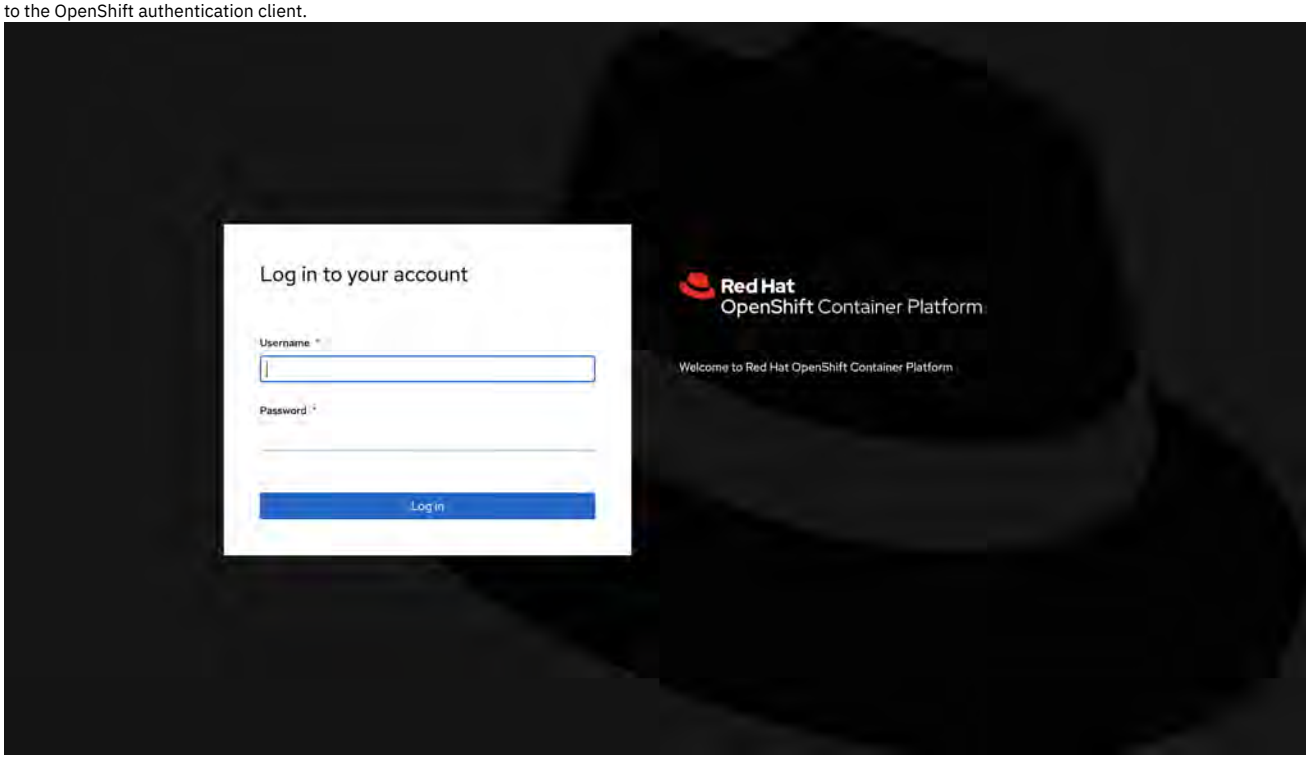
Log in to IBM Data Cataloging Dashboard with SSO enabled

Follow the steps to log in to the IBM Data Cataloging Dashboard:

1. Start the IBM Data Cataloging graphical user interface, and you get the logging screen validation as follows.



2. Wait for the validation for OpenShift Platform session. If you have an active session, you are redirected to Data Cataloging Dashboard, otherwise you are redirected to the OpenShift authentication client.



3. Enter your OpenShift credentials and click Log in button.
You are redirected to IBM Data Cataloging main dashboard.

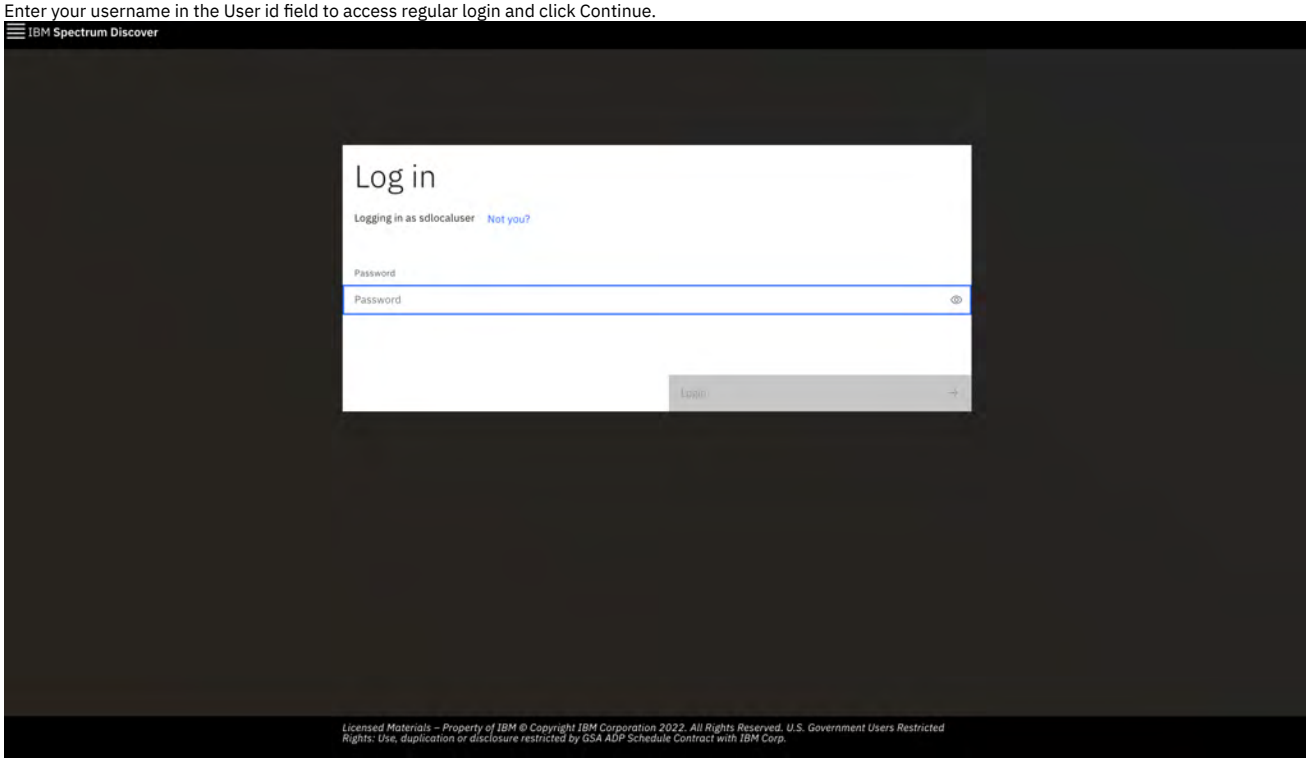


Note: Consider that IBM Data Cataloging for security renews the session every hour, and there are cases in which it is necessary to re-enter the OpenShift Container Platform credentials.

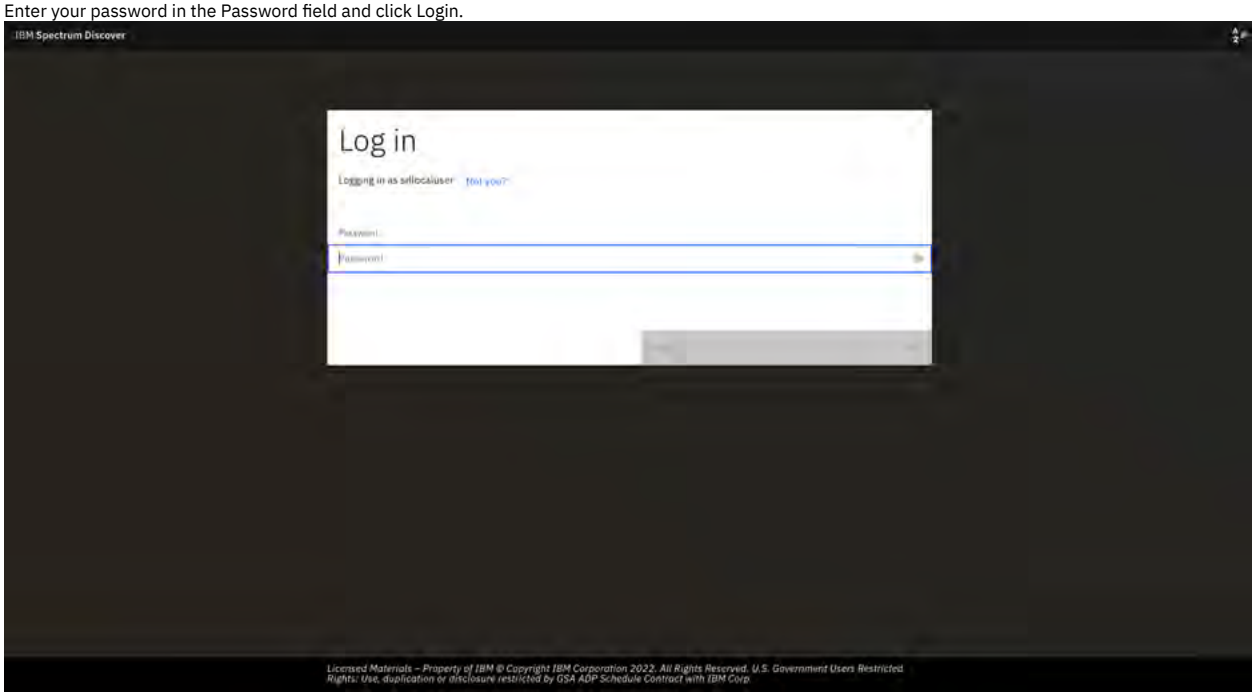
Log in to IBM Data Cataloging Dashboard with SSO disabled

If you see a previous user logged in, you need to click on the Not you? button to see the following screen.

1. Enter your username in the User id field to access regular login and click Continue.



2. Enter your password in the Password field and click Login.



3. After the credentials are validated, IBM Data Cataloging main dashboard is displayed as follows.



Note:
Regardless of which user appears in the User id field, clicking the Login with OpenShift Platform button will redirect you to the OpenShift authentication client and you can access using your OpenShift credentials.

Disabling SSO login

The single sign-on (SSO) enable by default when you install the IBM Data Cataloging service. Follow the steps to disable the SSO:

1. Login to OpenShift Container Platform web console.
2. Go to Operators > Installed Operators > and select the IBM Data Cataloging.
3. Under the `data-cataloging-service` instance, go to Details tab.
4. Scroll down to the Enabled Features section and clear the Single Sign-On checkbox.
5. Go to Workloads > Deployments and look for the `isd-ui-backend` deployment.
6. Go to YAML tab and change the `OCPAUTH_DEFAULT` envvar value from `True` to `False`.
7. Click Save and wait until a pod is created.
8. After the pod is created, clean cookies, cache, and reload the IBM Data Cataloging GUI.

When SSO is disabled as the default authentication method, the following default login credentials can be used for the IBM Data Cataloging graphical user interface:

Username
`sdadmin`
Password
`Passw0rd`

Resetting the sdadmin password

If the `sdadmin` password changes and you forget the password, you can access the keystone container and run the **reset_sdadmin_details.sh** script to reset the password to the original password.

Procedure

1. Issue the following command to get the keystone pod name:

   ```
   c -n ibm-data-cataloging rsh deployment/isd-keystone
   ```

   The system returns the pod details similar to the following:

   ```
   Name                                          READY   STATUS    RESTARTS   AGE
   spectrum-discover-keystone-599cf9fb77-bpqc8   1/2     Running     0        3d21h
   ```

2. Go to the directory `cd /` and run the reset_sdadmin_details.sh script to reset the details or the original password.

   ```
   : ussuri
   : 127.0.0.1
   : 127.0.0.1
   : 3
   : RegionOne
   : keystone
   : default
   : spectrum-discover
   : sdadmin
   : Passw0rd
   : admin
   : http://127.0.0.1:5000/v3
   : http://127.0.0.1:5000/v3
   : http://127.0.0.1:5000/v3
   keystone-manage bootstrap --bootstrap-project-name spectrum-discover --bootstrap-username sdadmin --bootstrap-password
   Passw0rd --bootstrap-role-name admin --bootstrap-service-name keystone --bootstrap-region-id RegionOne --bootstrap-admin-
   url http://127.0.0.1:5000/v3 --bootstrap-public-url http://127.0.0.1:5000/v3 --bootstrap-internal-url
   http://127.0.0.1:5000/v3
   ```

Password policies

Data cataloging 2.0.2.1 introduces password policies for the local users who are configured in the default authentication domain.

The password policies that are introduced for all local user accounts, enhance their security.

Note:
Data cataloging does not enforce password policies for the user accounts that are imported to the Data cataloging authentication scheme. These policies include all user accounts imported from the external domains like LDAP or IBM® Cloud Object Store that are configured with Data cataloging. Any password policies that are configured for these external authentication providers (LDAP/IBM Cloud Object Store), would apply to the corresponding users from these authentication domains.

Password policies

Data cataloging local users must follow the password policies that are defined in the 2.0.2.1 release.

Password strength requirements

- Passwords must have a minimum length of seven characters.
- Passwords must contain at least one letter.
- Passwords must contain at least one digit.

Unique password history requirements

- Users must create a unique password each time the password is changed. The new password cannot be any of the last five passwords previously used.

Password expiration requirements

- The User password expires after 90 days from the time it is changed.

Password change requirements
: Data cataloging users with Admin roles (like the "sdadmin" user) can create a new user or reset the password of an existing user. However, this password expires when the user logs in for the first time and must be changed immediately.

Account lockout requirements
: A user account is locked out for 1 hour after five successive failed login attempts.

Password upgrade for existing users
: Data cataloging deployments, upgraded from versions earlier than 2.0.2.1, include the new password policies that are applied to local user accounts. Existing user accounts are also impacted in the following ways:

  - Existing users can continue to use their current passwords to log in to the system.
  - Passwords for existing user accounts expire only in the following situations:
    - Passwords expire when users change their password. In this scenario, the new password will expire after 90 days.
    - Passwords expire when the administrative user resets the user password. In this scenario, the updated password expires immediately after the first login and the user must create a new unique password.
  - When the user password is changed, the following password policies are enforced:
    - **Password strength requirements**
    - **Unique Password history requirements** - This policy restricts users from reusing any of the last five passwords.
  - On completing the product upgrade, the **Account lockout requirements** policy is immediately enforced for all local users that includes all existing users.

  Note: To apply all the password policies to the local user accounts after they upgrade to 2.0.2.1 or later releases, follow the listed recommendations:

  - The Admin user resets passwords for all the existing local user accounts and communicates the new password to the respective users.
  - All the local users use the UI Password change REST API to change their passwords. For more information, see Changing password and /auth/v1/users/<user_ID>/password: POST */auth/vi/users/<user_ID>/password: Post* in *IBM Data Cataloging: REST API Guide*.

Changing password

To change the password, you need to provide the existing and new password.

About this task

To change your password, follow these steps.

Procedure

1. Click the user icon.
2. Under Account Settings, click Change Password.
3. Enter the existing password, new password, and new password confirmation.
4. Click Save.
   Note: When the password is changed successfully, you are logged out of IBM Data Cataloging. You need to relogin with the new password.

Managing user accounts

The administrator can add and manage local user accounts, which are authenticated by IBM Data Cataloging. The administrator can also assign local or LDAP and IBM Cloud® Object Storage-managed users to roles and collections.

Use the Users tab on the Access page to view information about user accounts that are authenticated by the local domain or either an LDAP or IBM Cloud Object Storage server. You can also use the tab to create, edit, or delete local users. You cannot create or delete either LDAP or IBM Cloud Object Storage user accounts, but you can assign these users to roles and collections.

Adding a local user account
: To add a local user account that is authenticated by IBM Data Cataloging, click Create new user and the Add local User window opens. For more information, see Creating user accounts.

Editing a user account
: You can edit account information for a local user. You cannot edit the details of either Lightweight Directory Access Protocol (LDAP) or IBM Cloud Object Storage user accounts, but you can assign these users to roles and collections.
To edit a local user account, select the user that you want to edit and click Edit. Use the Edit User window to edit the local user account.

: To edit an LDAP or IBM Cloud Object Storage user account roles, select the user that you want to edit and click Edit. Use the Edit User window to assign these users to roles and collections.

Deleting a local user account
: To delete a local user account, select the user that you want to delete and click Delete.

User information

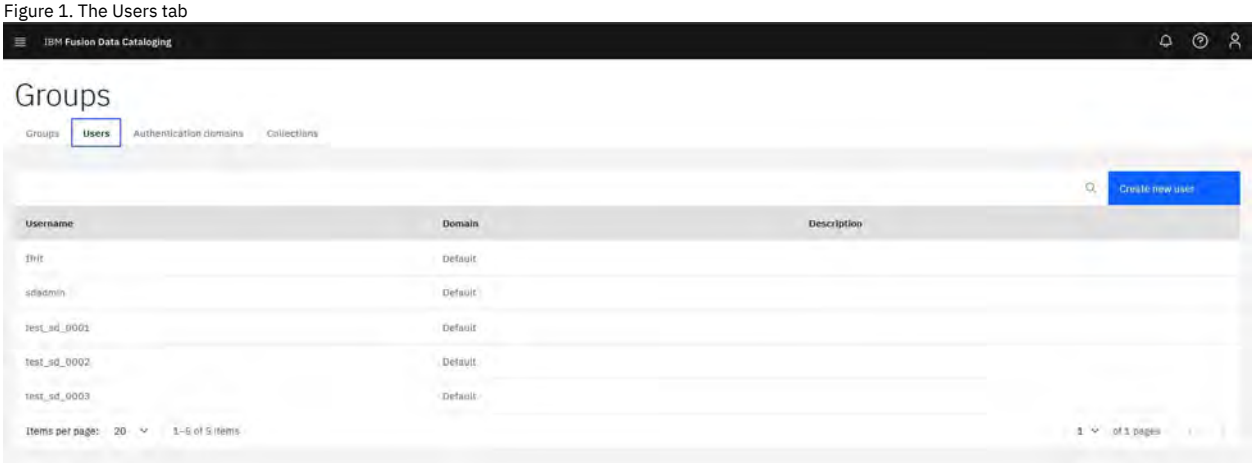
The Users tab lists the users that are available from the local domain and from either LDAP or IBM Cloud Object Storage connections. The tab includes the following user account information:

User Name
: Indicates the username for the account.

Domain
: Indicates the domain that provides authentication for the user. For authentication by IBM Data Cataloging, the domain name is Default.

Description
: Indicates the description of the user.

Figure 1. The Users tab



- **Creating user accounts**
  The administrator can add local user accounts, which are authenticated by IBM Data Cataloging, and assign roles to users.

Creating user accounts

The administrator can add local user accounts, which are authenticated by IBM Data Cataloging, and assign roles to users.

About this task

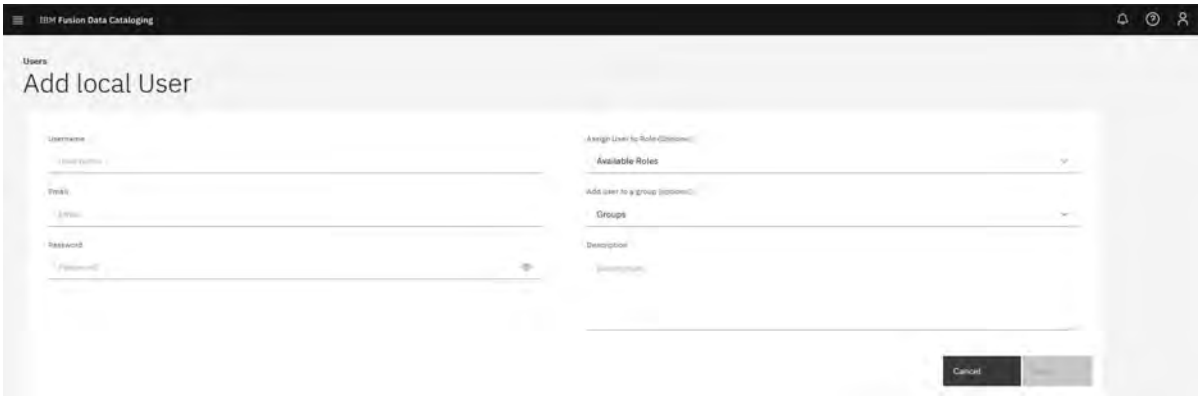
Go to the Users tab on the Access management page to add a local account.

- You can also assign roles and passwords to users.
- You can also add a user to a group.

Procedure

1. Log in to the IBM Data Cataloging web interface.
2. Click menu and go to Access management > Users. Click Create new users to open Add local User window.

   Figure 1. The Add local User window



3. Enter a User Name and Email address for the user.
4. Enter a Password for the user.
5. Choose the available role from the Assign User to Role list to assign one or more roles to the user.
   Note:
   - This step is optional. For more information about roles, see Roles.
   - Users that are assigned the Data User or Collection Admin role must also be associated with at least one collection.
6. Use the Assign User to Group list to assign the user to one or more user groups.
   Note: This step is optional. You can also use the Groups tab to assign users to groups.
7. Provide description about user in the Description field.
   Note: This step is optional.
8. Click Save.

Managing groups

The administrator can create and manage local groups that are authenticated by IBM Data Cataloging. The administrator can also assign local or Lightweight Directory Access Protocol (LDAP) and IBM Cloud® Object Storage system-managed groups to roles and collections.

Use the Groups tab on the Access management page to view information about groups accounts that are authenticated by either a local domain, an LDAP server, or the IBM Cloud Object Storage server. You can also use the tab to add, edit, or delete local groups. You cannot edit or delete LDAP or IBM Cloud Object Storage groups, but you can assign these groups to roles and collections.

Adding a local group
: To add a local group, click Create new group to open the Add Local Group window. For more information, see Creating groups.

Editing a group
: To edit a group, select the group that you want to edit and click Edit. Use the Edit Group window to edit the local group.

Deleting a local group
: To delete a local group, select the group that you want to delete and click Delete.

Group information

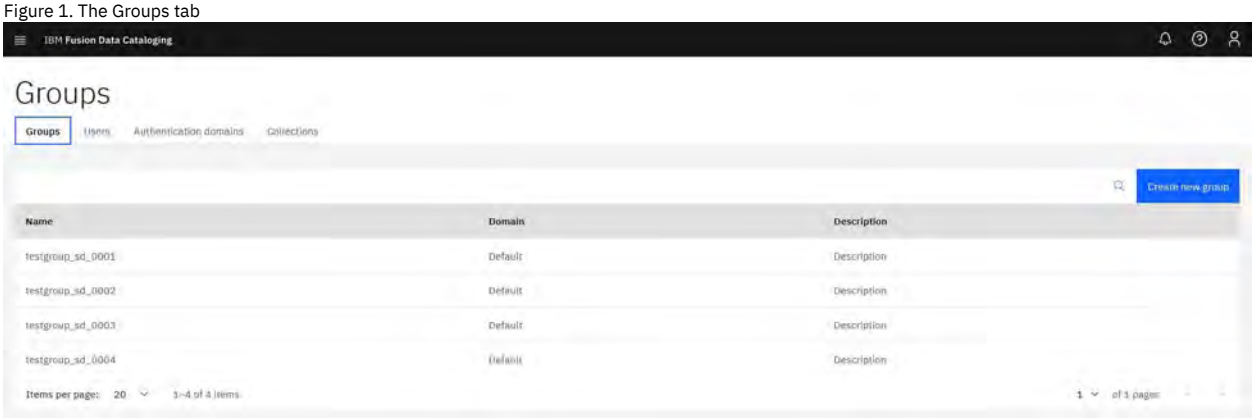
The Groups tab includes the following information:

Group Name
: Indicates the name for the group.

Domain
: Indicates the domain that provides authentication for the group. For authentication by IBM Data Cataloging, the domain name is Default.

Description
: Indicates the description of the group.

Figure 1. The Groups tab



- **Creating groups**
  The administrator can add local groups that are authenticated by IBM Data Cataloging and assign users and roles to the groups.

Creating groups

The administrator can add local groups that are authenticated by IBM Data Cataloging and assign users and roles to the groups.

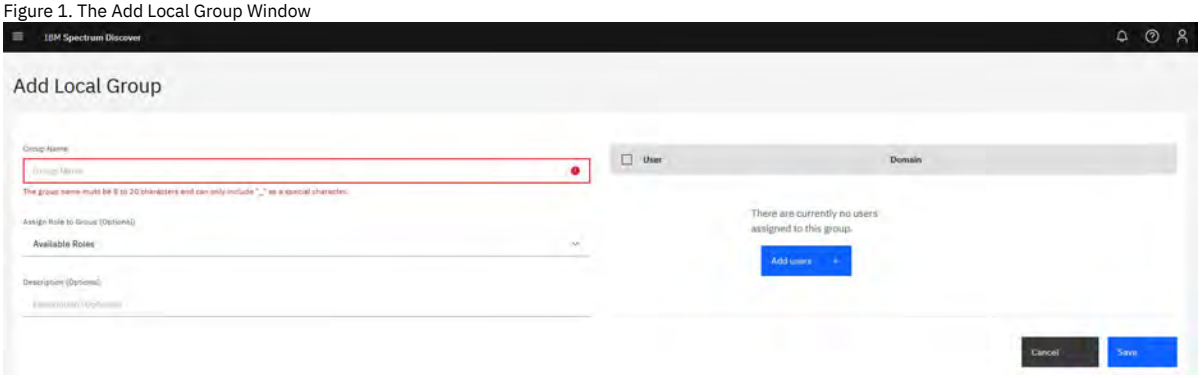
About this task

Go to the Groups tab on the Access management page to add local groups. You can assign users and roles to the group and add the group to a collection.

Procedure

1. From the Groups tab of the Access management page, click Create new group to open the Add Local Group window.

   Figure 1. The Add Local Group Window

2. Enter a Group Name.
3. Use the Assign Role to Group list to assign one or more roles to the group.
   Note:
   - This field is optional.
   - For more information, see Roles.
   - Groups that are assigned the Data User role or Collection Admin role must be associated with at least one collection.
4. Click Add Users to open the Add Users window and add one or more local users to the collection.
   a. Enter a username that you want to add to the group and press Enter. The window lists each name that you enter. Click a name to remove it from the list. Click Add to add the users to the group.
   The Users list displays the following details for users that are added to the group.

   Username
   : The username or email address of the member.

   Domain
   : The domain that provides authentication for the member.

5. Enter the group detail in the Description field for the group.
   Note: This step is optional.
6. Click Save.

Managing collections

Collections are logical groups of metadata that share a common member access list. For example, a collection can restrict metadata within a research project to the members of the project only. Members outside of the project cannot see the metadata.

The administrator can:

- Create collections.
- Assign users and groups to collections.
- Create a policy to associate specific metadata that is collected by IBM Data Cataloging with the collection.
- Assign a collection to a connection to associate specific metadata from that connection data source that is collected by IBM Data Cataloging with the collection.

Users with the Data Admin role can view all metadata that is collected by IBM Data Cataloging and are not restricted by collections. Users with the Data Admin role can create policies that assign a collection value to a set of records, thus grouping a set of records under a collection.

Users with the Collection Admin role can add the Data User role to user for collections that they administer. Adding this role gives data users access to a particular collection, which allows the users to access the records that are marked with that collection value.

Collection User
: Users with this role can access metadata that is collected by IBM Data Cataloging, but metadata access can be restricted by the collections that are assigned to users in this role.

  - Users assigned with the Collection User role can:
    - Run scans of collections the user is assigned.
    - View policies of the collections the user is assigned.
    - List any type of tag.
  - Users assigned with the Collection User role cannot:

- Create, update, and delete any policies.
- Create, modify, and delete any tags.

Use the Collections page to manage collections.

Creating a collection
: To create a collection, click Create Collection. For more information, see Creating collections.

Editing a collection
: To edit a collection, select the collection that you want to edit and click Edit Collection. Use the Edit Collection window to edit the collection.

Deleting a collection
: To delete a collection, select the collection that you want to edit and click Delete Collection.

Collections information

The Collections page includes the following information:

Collection Name
: Indicates the name of the collection.

Description
: Indicates the description of the collection.

Figure 1. The Collections tab



- **Creating collections**
  The administrator can create collections or assign users and groups to collections. A Collection Admin administrator can assign users and groups only to collections that they administer. A Data Admin administrator can use the AUTOTAG policy to associate metadata records with a collection.

Creating collections

The administrator can create collections or assign users and groups to collections. A Collection Admin administrator can assign users and groups only to collections that they administer. A Data Admin administrator can use the AUTOTAG policy to associate metadata records with a collection.

About this task

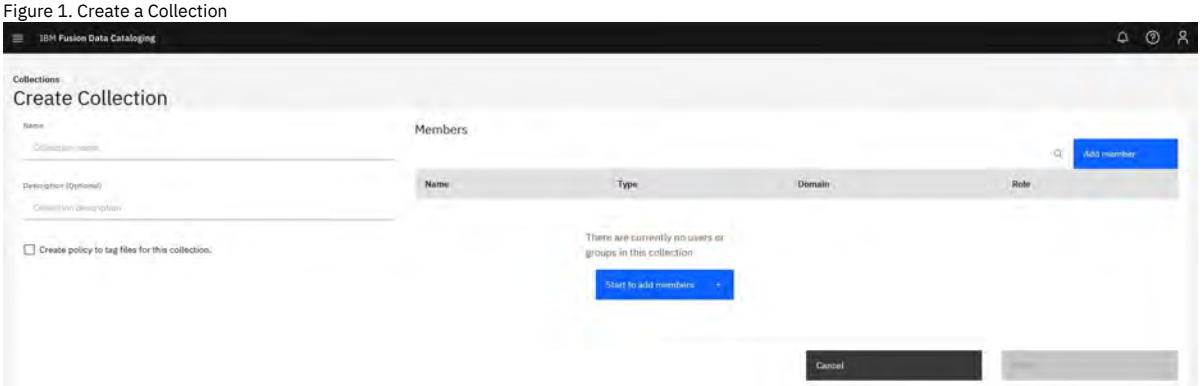
Collections are logical groups of records. Access to these record groups is restricted to specific users or groups. The administrator can associate policies with an appropriate collection value so that searches can be restricted to only the scope that a user or group has permissions to see.

Use the Collections tab on the Access management page to create collections.

Procedure

1. From the Collections tab of the Access management page, click Create collection to open the Create Collection window.

   Figure 1. Create a Collection

2. Enter a collection Name and the details in the Description field.
   Note: The Description field is optional.
3. Click Start to Add Members to open the Add Members window and add one or more users or groups to the collection.
   a. Enter a username, group name, or email address of a member to include in the collection and press Enter. The window lists each name or address that you enter. Click a name or address to remove it from the list. Click Add member to add the members to the collection.
   b. Select the role for the member on the collection. The default role is Data User.

The Members area lists the following details for the members of the collection.

Name
: The username, group name, or email address of the member.

Type
: The account type: user or group.

Domain
: The domain that provides authentication for the member.

Role
: The role on the collection that is assigned to the member.

4. To create a policy for the collection, select Create policy to tag files for this collection. For more information, see Managing metadata policies.
5. Click Create.

Managing LDAP and IBM Cloud Object Storage System connections

The administrator can create and manage connections to LDAP or IBM Cloud® Object Storage System servers that provide authentication for IBM Data Cataloging users.

Use the Authentication Domains tab on the Access page to create, test, manage, or delete LDAP connections.

You can create a connection that includes all users and groups that are authenticated by an LDAP server or only users or groups within a specified LDAP member range.
Note: You cannot specify a member range for users and groups that are managed by the IBM Cloud Object Storage System.

Creating a connection
: To create a connection to an authentication domain, click Add Domain Connection.
For steps to create a connection to an LDAP server, see Creating an LDAP connection.

  For steps to create a connection to an IBM Cloud Object Storage system server, see Creating an IBM Cloud Object Storage connection.

Editing a connection
: To edit a connection, click Edit.

Deleting a connection
: To delete a connection, click Delete.

Note: You cannot map the LDAP roles from your connection and must use an admin user to make the role assignment because the roles between IBM Data Cataloging and your LDAP server might be different.

- **Creating an LDAP connection**
  The administrator can create a connection to a Lightweight Directory Access Protocol (LDAP) server that provides authentication for IBM Data Cataloging users.
- **Creating an IBM Cloud Object Storage connection**
  The administrator can create a connection to an IBM Cloud Object Storage server that provides authentication for IBM Spectrum Discover users and groups from the corresponding domain.

Creating an LDAP connection

The administrator can create a connection to a Lightweight Directory Access Protocol (LDAP) server that provides authentication for IBM Data Cataloging users.

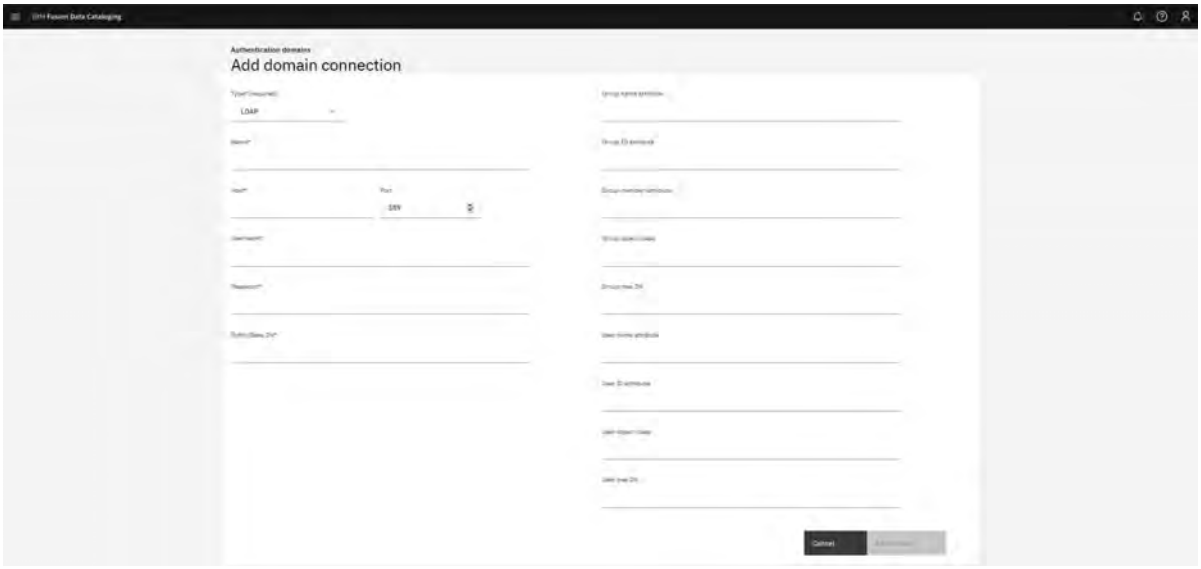
About this task

Use the Authentication Domains tab on the Access page to create an LDAP or secure LDAP (LDAPS) connection. You can create a connection that includes all users and groups that are authenticated by an LDAP server or only users or groups within a specified LDAP member range.
Note: You cannot map the LDAP roles from your connection and must use an admin user to make the role assignment because the roles between IBM Data Cataloging and your LDAP server might be different.

Procedure

1. From the Authentication Domains tab of the Access page, click Add Domain Connection to open the Add Domain Connection window.
2. From the Type list, select LDAP or LDAPS.

Figure 1. Create an LDAP connection



3. Enter the following information for the LDAP directory:

   Name
   : Indicates a name that IBM Data Cataloging associates with the connection to the directory that provides authentication.

   Port
   : Indicates the LDAP server port that provides the connection.

   Username
   : Indicates the distinguished name (DN) for the user that is used to access directory name entries. Use the following format:

     ```
     cn=relative_distinguished_name dc=domain_component
     ```

     For example,

     ```
     cn=Randy Marsh,dc=example,dc=com
     ```

   Password
   : Indicates the password for the user name.

   Suffix/Base DN
   : Indicates the DN that is the base of entry searches in the directory. For example:
     - `dc=test`
     - `dc=org`

   Group Name Attribute
   : Indicates the LDAP attribute that is mapped to the group name.

   Group ID Attribute
   : Indicates the LDAP attribute that is mapped to the group ID.

   Group Member Attribute
   : Indicates the LDAP attribute that is mapped to show group membership.

   Group Object Class
   : Indicates the LDAP object class for groups.

   Group Tree DN
   : Indicates the DN that is the base for group searches.

   Username Attribute
   : Indicates the LDAP attribute that is mapped to the user name.

   User ID Attribute
   : Indicates the LDAP attribute that is mapped to the user ID.

   User Object Class
   : Indicates the LDAP object class for users.

   User Tree DN
   : Indicates the DN that is the base for user searches.

4. Click Connect.

- **Configuring self-signed certificates to secure LDAPS connection**
  Establish secure LDAPS connections with self-signed certificates to protect the confidential data.

Configuring self-signed certificates to secure LDAPS connection

Establish secure LDAPS connections with self-signed certificates to protect the confidential data.

About this task

IBM Data Cataloging supports LDAPS domain connections with LDAP servers that are deployed with trusted (CA signed) certificates. To use self-signed certificates, you need to add the self-signed certificate to the IBM Data Cataloging keystone pod trusted certificates list.

Procedure

1. Ensure that the cacerts directory exists in the following path, which is accessible to the keystone pod:

   ```
   /opt/ibm/metaocean/data/keystone/cacerts/
   ```

   If the directory is not available, create it by running the following command:

   ```
   oc -n ibm-data-cataloging exec deployment/isd-keystone -- mkdir -p /opt/ibm/metaocean/data/keystone/cacerts
   ```

   Then, retrieve the keystone pod name and copy the self-signed certificate to the directory:

   ```
   KEYSTONE_POD=$(oc -n ibm-data-cataloging get pod -o jsonpath='{.items[0].metadata.name}' -l role=keystone)
   oc -n ibm-data-cataloging cp cert.crt $KEYSTONE_POD:/opt/ibm/metaocean/data/keystone/cacerts/cert.crt
   ```

2. Update the certificates within the keystone pod by running the following command:

   ```
   oc -n ibm-data-cataloging exec deployment/isd-keystone /update-cacerts.sh
   ```

3. Restart the keystone pod to apply the changes by executing:

   ```
   oc -n ibm-data-cataloging delete pod -l role=keystone
   ```

Creating an IBM Cloud Object Storage connection

The administrator can create a connection to an IBM Cloud® Object Storage server that provides authentication for IBM Spectrum® Discover users and groups from the corresponding domain.

About this task

Use the Authentication Domains tab on the Access page to create a connection to an IBM® Cloud Object Storage System server. You must provide credentials for the IBM Cloud Object Storage security administrator.

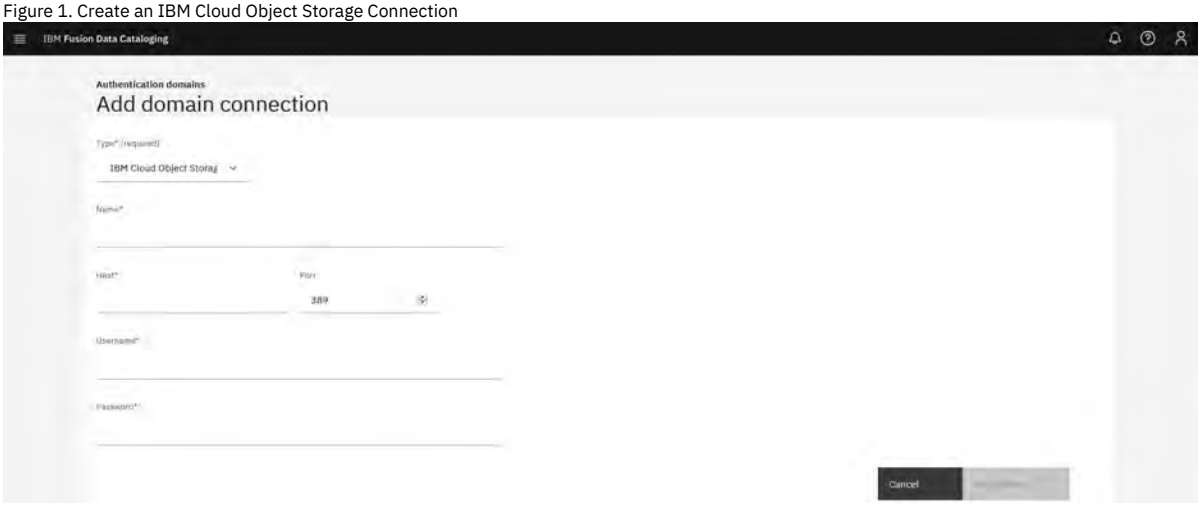
All users and groups that are managed by the IBM Cloud Object Storage are available for IBM Data Cataloging. You cannot specify a member range for these connections.

Note: You cannot run scans unless you add override warnings in the configuration file.

Procedure

1. From the Authentication Domains tab of the Access page, click Add Domain Connection to open the Add Domain Connection window.
2. From the Type list, select IBM Cloud Object Storage.

Figure 1. Create an IBM Cloud Object Storage Connection



3. Enter the following information for the IBM Cloud Object Storage connection:

   Name
   : Indicates the IBM Cloud Object Storage domain name.

   Host
   : Indicates the IBM Cloud Object Storage hostname.

   Port
   : Indicates the IBM Cloud Object Storage port number.

   User name
   : Indicates the IBM Cloud Object Storage security administrator name.

   Password
   : Indicates the IBM Cloud Object Storage security administrator password.

4. Click Connect.

Managing metadata policies

Policies might be used to automatically tag a set of documents on a periodic basis. In addition, policies might be used to send sets of documents to be deep-inspected by a registered application.

Roles and permissions

Data User
: Users with this role can create, modify, and view their policies. Policies can be applied only to the collections the user has access to.
Users with the Data User role cannot use a **`COLLECTION`** tag when they create or modify policies.

Collection Admin
: Users with this role can create, modify, and view their policies. Policies can be applied only to the collections that they administer. Users with the Collection Admin role cannot use a **`COLLECTION`** tag when they create or modify policies.

Collection User
: Users with this role can access metadata that is collected by IBM Data Cataloging, but metadata access can be restricted by the collections that are assigned to users in this role.

  - Users assigned with the Collection User role can:
    - Run scans of collections the user is assigned.
    - View policies of the collections the user is assigned.
    - List any type of tag.
  - Users assigned with the Collection User role cannot:
    - Create, update, and delete any policies.
    - Create, modify, and delete any tags.

Data Administrator
: Users with this role can create, modify, view, and delete policies.

Security Administrators
: Users with this role cannot create, modify, view, or delete policies.

Service User
: Users with this role cannot create, modify, view, or delete policies.

- **Adding policies**
  You can add policies to help with data management.
- **Running policies**
  You can configure policies to run at specified or scheduled times, at policy creation time, or when the system is manually started, paused, restarted, or stopped.
- **Viewing policies**
  You can view your policy information.
- **Modifying policies**
  You can modify your policies to work with and use your data more effectively.
- **Deleting policies**
  You can delete policy information.

Adding policies

You can add policies to help with data management.

About this task

To add a policy, you must define the policy and its parameters, establish a schedule to run the policy, and save the policy.
You can add custom metadata values to all or a subset of the records based on filter criteria. For example, you can add a project name to records based on their location within the file system or owner ID.

You can add a policy filter, which is similar to the **where** clause in an SQL query. The filter must be constructed by using standard SQL syntax:

- To enact a policy on all files not accessed in one year, the filter might be written as:

  ```
  atime < (NOW() - 365 DAYS)
  ```

- To enact a policy on all files owned by **`Smithers`**, the filter might be written as:

  ```
  owner='Smithers'
  ```

- To enact a policy on all PDF files in cluster **`cl1`** and data source **`fs1`**, the filter might be written as:

  ```
  cluster='cl1' and datasource='fs1' and filetype='pdf'
  ```

Procedure

1. Go to Metadata > Tag Management.
2. Under Policies click Add Policy.
3. Enter a name for the policy in the Name box (for example, `MyPolicy`).
4. Select the required policy type from the Policy Type menu.
   The policy types are:

- AUTOTAG
- CONTENT SEARCH
- DEEP-INSPECT

5. Click Next step.
6. Complete the policy information:
   a. Select the list of Collections the policy applies to. If no collections are selected, the policy applies to all collections available to the user when run.
   b. You must use a filter for the policy. The filter defines which set of records the policy acts on. Enter your filter into the Filter box (for example, `size > 100`).
   c. Complete the policy-specific parameters based on which policy type you select in step 3:
      i. You can set parameters for **AUTOTAG** policy types. For more information, see Adding auto-tagging policy parameters.
      ii. You can set parameters for **DEEP-INSPECT** policy types. For more information, see Adding deep-inspection policy parameters.
      iii. You can set parameters for **CONTENT SEARCH** policy types. For more information, see Adding content search policy parameters.
7. Click Next step
8. Now that your policy parameters are defined, you must schedule the frequency.
9. Click the slider control to set the status to one of the following values:

   Active
   An *Active* policy is run whenever its scheduling event is reached.
   Inactive
   An *Inactive* policy is not run when its scheduling event is reached, including the **Now** event.

10. Select a Schedule to apply the policy. Indicate whether you want to schedule the policy `Now`, `Daily`, `Weekly`, or `Monthly`.
    Note: Policy schedule times are entered in Coordinated Universal Time (UTC).

    Now
    Indicates that the policy is applied immediately unless the policy's status is **Inactive**.
    Daily
    Indicates a specific time of day to apply the policy. Enter the time of day by clicking the hour and minute from the widget that is shown. The policy is applied daily at the specified time.
    Weekly
    Indicates a specific day and time in which to apply the weekly policy:
    a. Enter the time of day to apply the policy by clicking the hour and minute from the widget that is shown.
    b. Select the day of the week from the list of days.
    The policy is applied once a week on the specified day and time.
    Monthly
    Indicates a specific day and time in which to apply the monthly policy:
    a. Enter the time of day to apply the policy by clicking the hour and minute from the widget that is shown.
    b. Select the date by clicking the month and day from the widget that is shown.
    The policy is applied once a month on the specified day and time.

11. Click Next step.
12. Review the data and click Save to save the new policy.
    The new policy displays in the list of policies under the Policies tab.

- **Adding auto-tagging policy parameters**
  You can add specific parameters for auto-tagging policies after you define the policy type and set up the mandatory filter or optional collection information.
- **Adding deep-inspection policy parameters**
  You can add DEEP-INSPECT policies.
- **Adding content search policy parameters**
  You can add search content parameters for your policies.

Adding auto-tagging policy parameters

You can add specific parameters for auto-tagging policies after you define the policy type and set up the mandatory filter or optional collection information.

Procedure

1. Associate one or more tags with this policy by using one of the following methods:
   a. Click +Add Tag.
   b. Select a tag name from the Field menu.
   The Fields can also be specified by going to Metadata > Tags.
   c. Select which Tag to apply (for example, `TEMPERATURE`) and then enter the appropriate Values.
   Note: If you do not know the valid values for a tag, go to Search in the main menu and select the tag from the Start a visual exploration list. Click the Go "circle arrow" icon. The valid values of the tag are displayed.
   d. To delete a Field, click the Delete "minus" icon next to a field.
   e. You can add more tag values by clicking the +Add Tag control.
   Each new Field defaults to the next item in the Field menu.
   Or you can automatically extract the tag from the directory path by clicking the Extract tag from box. For example, you might have files in a directory structure by department and want to extract the department name into a tag. Automatically extract the tag:
      i. Select the Extract tag from path checkbox.
      ii. Select a tag from the Field menu.
      iii. Specify the Depth in the path to be used as the value of the tag.
   To create a new tag, see Creating tags.
2. Click Next Step.

Adding deep-inspection policy parameters

You can add DEEP-INSPECT policies.

About this task

You can enrich metadata through an external deep inspection application. For more information, see Managing applications. For example, you can extract patient names from medical records and index the names. Indexing the names helps when you search for files that pertain to patients by name. Deep inspection policies can send lists of files to an application, which can examine the contents of files and return the values that it finds paired with defined tag keys.

Important: If the deep-inspect application returns a **`tag:value`** pair that is not requested by the deep-inspect policy, the tag:value pair is still updated by the deep-inspect policy as long as the tag resides in IBM Data Cataloging.

Procedure

Do the following to add parameters for deep-inspection policies:

1. Add the Application name (for example, `example-application`).
   Note: To add an application, see /policyengine/v1/applications: POST*/policyengine/v1/applications: POST* in the *IBM Data Cataloging: REST API Guide*.
2. Click +Add Tag.
3. Select which Parameter to apply (for example, `extract-tags`), and then select the appropriate Values (for example, `TEMPERATURE`).
4. You can add more parameters by clicking the +Add parameter control. This is optional.
5. You can delete a parameter by clicking the Delete "minus" icon next to the parameter's Value. This is optional.
6. Click Next Step.

Adding content search policy parameters

You can add search content parameters for your policies.

About this task

You can enrich metadata through the built-in content search application. For more information, see Using content search policies.

You can add specific parameters for content search policies by using the following steps:

Procedure

1. Select `contentsearchagent` for the Application.
2. Click +Add Row to open the Parameter dialog.
3. Enter a tag name in the Parameter box and select one or more Search Expressions from the dropdown list.
4. For the Value, select either True/False or Value matching expression.
5. Repeat steps 1 - 3 to set other tags that use the same policy.
6. Click Next Step.

Running policies

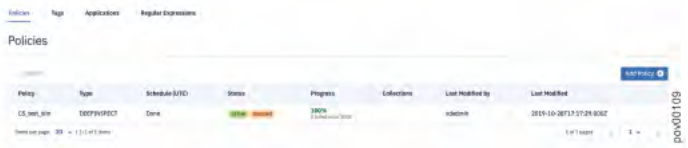
You can configure policies to run at specified or scheduled times, at policy creation time, or when the system is manually started, paused, restarted, or stopped.

Procedure

To configure a policy to run:

1. Go to Metadata
2. Click a policy to select it. The following screen displays:

   Figure 1. Policies table



From the screen, you can perform the following actions on the selected policy:

Pause
: Click the "vertical bars" icon to pause a policy that is running. When a policy is paused, it enters a `Paused` state. A policy cannot be paused until the current batch that is being processed finishes.

Start
: Click the "right-arrow" icon to resume a paused policy or to start a stopped policy from its beginning. When a policy is started, it enters a `Running` state.

Stop
: Click the "square" icon to stop a policy that is in progress. When a policy is stopped, it enters a `Stopped` state.

When a policy completes, it enters a `Stopped` state. The progress column indicates the success or failure status of the policy. If there are failures, you can examine the per policy execution log files to obtain more details of the failures.

For more information, see Viewing policies.

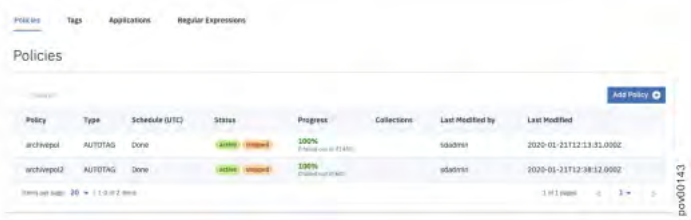
Viewing policies

You can view your policy information.

About this task

You can see a list of the active and inactive policies and their status.

Figure 1. Policies table



Procedure

1. Go to Metadata > Tag Management
2. Under Policies, view a table of the specified policies with the following aspects:

   Policy
   : Displays the name of a policy.

   Type
   : Displays the policy type. There are three types:
   - **AUTOTAG**: You can apply custom metadata values to some, or all of the records based on filter criteria.
   - **CONTENT SEARCH**: You can enrich metadata by using the built-in CONTENTSEARCH application.
   - **DEEP-INSPECT**: You can enrich metadata through content inspection of source data.

   Schedule
   : Displays the frequency at which a policy is applied. Policy schedule times are displayed in Coordinated Universal Time (UTC).

     Done
     : Indicates that the policy is applied.

     Daily: 00:00
     : Indicates that the policy is applied one time a week on the displayed day and time.

     Weekly:[day], 00:00
     : Indicates that the policy is applied one time a week on the displayed day and time.

     Monthly:[date], 00:00
     : Indicates that the policy is applied one time a month on the displayed date and time.

   Status
   : Displays the status and current state of the policy.
   - A policy's **Inactive** status shows the **None** state.
   - A policy's **Active** status can have a state of **Initialized**, **Running**, **Paused**, or **Stopped**.

   Progress
   : Displays the percentage of completion of a policy.
   If there is an Error "yellow triangle" icon, you can hover the cursor over it to see more information.

     If there are some records that met the filter criteria but fail to be tagged, the number of failed records and the total number of records are displayed below the percentage of completion.

     If there are failures, the per policy execution log files can be examined to obtain more details of the failures. For more information, see Viewing policy history details and logs.

   Collections
   : Displays the name of the collection that the policy applies to or the number of collections the policy applies to if there is more than one collection.

   Last modified by
   : Displays the name of the user who last modified the collection.

   Last modified
   : Displays the date and time that the collection was last modified.

   You can add a policy by clicking Add Policy +. For more information, see Adding auto-tagging policy parameters or Adding deep-inspection policy parameters on page 19.
3. Click a policy to select it. The following screen displays:

   Figure 2. Policies table



From the screen, you can perform the following actions:

Start, Pause, Stop, or delete
: You can start, pause or stop a policy. For more information, see Running policies.

Edit/Delete
: Use the Edit "pencil" icon to modify a policy. Use the Delete "trashcan" icon to delete a policy. You cannot delete a policy that is running (The "trashcan" icon is made unavailable).

View
: Preview the details of the selected policy.

- **Viewing policy history details and logs**
  You can view the history details, execution statistics and debug log data for all older or current metadata policies.

Viewing policy history details and logs

You can view the history details, execution statistics and debug log data for all older or current metadata policies.

About this task

Follow the procedure to view policy history details and logs.

Procedure

1. Go to Metadata > Tag Management.
2. Under Policy History, view the following information for the policy that ran recently or was run in the past.

   Policy name
   : The name of the policy that was run.

   Type
   : The policy type that was run.

   Started
   : The time when the policy execution started.

   Ended
   : The time when the policy execution was completed. This column remains blank if the policy is running.

   Status
   : The current or final status of the policy execution.

   Total records
   : The number of indexed records that matched the policy filter criteria for this policy execution.

   Successful
   : The number of records that were processed successfully by the policy execution.

   Skipped
   : The number of records that matched the filter criteria, but were skipped by the policy execution.

   Failed
   : The number of records that the application failed to process.

   Note: Click the table header for each column to sort the tables. You can sort tables in both ascending or descending order.
3. Click to select a row in the policy history table.
4. Click View log to view the run log contents for the selected policy execution.
   Note: The run log also displays the details of all failed records. You can download and debug the logs if necessary.

Modifying policies

You can modify your policies to work with and use your data more effectively.

About this task

You can modify a policy from the table on the Policies page. You cannot change the Name or Type of a policy.

Procedure

1. Go to Metadata > Tag management
2. Under Policies, select a policy you want to modify and click the Edit "pencil" icon on the header.
   The Edit policy page appears. The policy parameters displayed depends on the policy type. The policy configuration area changes depending on the Extract tag from path setting.

Note: You cannot modify the name and the policy type.

3. Modify the collections that the policy applies to (if required).
4. Modify or enter a filter in the Filter box.
   The filter must be constructed by using standard SQL syntax. For more information, see example filters on page 17.
5. Modify the parameters specific to the policy type (if required).
   - For more information about `AUTOTAG` policy types, see Adding auto-tagging policy parameters.
   - For more information about `DEEP-INSPECT` policy types, see Adding deep-inspection policy parameters.
   - For more information about `CONTENT SEARCH` policy types, see Adding content search policy parameters, Adding deep-inspection policy parameters, or Managing user accounts.
6. Click the slider control to set the status to one of the following values:

   Active
   An *Active* policy is run whenever its scheduling event is reached.
   Inactive
   An *Inactive* policy is not run when its scheduling event is reached, including the **Now** event.

7. Specify a Schedule to apply the policy. Policy schedule times are entered in Coordinated Universal Time (UTC).

   Now
   The policy is applied immediately, unless the policy's status is **Inactive**
   Daily
   Indicates a specific time of day to apply the policy. Enter the time of day to apply the policy by clicking the hour and minute from the widget that displays.
   The policy is applied daily at the specified time.
   Weekly
   Indicates a specific day and time in which to apply the weekly policy:
   a. Enter the time of day to apply the policy by clicking the hour and minute from the widget that displays.
   b. Select the day of the week from the list of days.
   The policy is applied one time a week on the specified day and time.
   Monthly
   Indicates a specific day and time in which to apply the monthly policy:
   a. Enter the time of day to apply the policy by clicking the hour and minute from the widget that displays.
   b. Select the date by clicking the month and day from the widget that displays.
   The policy is applied one time a month on the specified day and time.

8. Under Review check all the changes you have made.
9. Click Save to save the changes.
   The modified policy is displayed in the list of policies on the Policies tab.

Deleting policies

You can delete policy information.

About this task

A policy can be deleted from the table on the Policies page. You cannot delete a policy that is running. A user with the role Data User can delete only their own policies.

Procedure

1. Go to Metadata > Policies.
2. Click the Delete "trashcan" icon in the Edit/Delete column of the policy you want to delete.
   If the "trashcan" icon is unavailable, then the policy is not available for deletion.
3. Click Delete in the confirmation window.
   The policy is removed from the table in the Policies tab.

Using content search policies

You can define regular expressions to search for and create policies that use the regular expressions.

About this task

You can enrich metadata through content inspection of source data by using the built-in CONTENTSEARCH application. To use this function, you can define regular expressions to search for and create policies that use these regular expressions.

When the policy runs, the documents are retrieved from the source system by the CONTENTSEARCH application, converted to text format if necessary, and searched by using the defined regular expressions. The results of the search are returned to Data cataloging and the metadata of the files that are updated. To create a CONTENT SEARCH policy, see Adding policies.

When you create a CONTENT SEARCH policy, you can select:

- Any tag type (including `Open`, `Restricted`, and `Characteristics` tags) for the `CONTENTSEARCH` application.
- Any regular expression.
- Either `"True/False"` or `"Value matching expression"`. `"True/False"` sets the tag value to either `True` or `False` if a match is found or not found. `"Value matching expression"` sets the tag value to the extracted content match.

Remember:

- If you select a **Restricted** tag with a defined set of values and choose to extract the value from a document, the value that is extracted must match one of the **RESTRICTED** values in the tag.
- If the content extracted exceeds the minimum tag value length, the extracted content is truncated.

- **Identifying the required regex expressions**
  The following information can help you identify required regex expressions.
- **Creating a content search policy**
  You can create a content search policy to filter a search query that finds candidate documents to which you can apply the policy.
- **Viewing content search application logs**
  The following information describes how to view the content search application logs.
- **Hints and tips for using content search**
  Following are some best practices for using the content search feature.

Identifying the required regex expressions

The following information can help you identify required regex expressions.

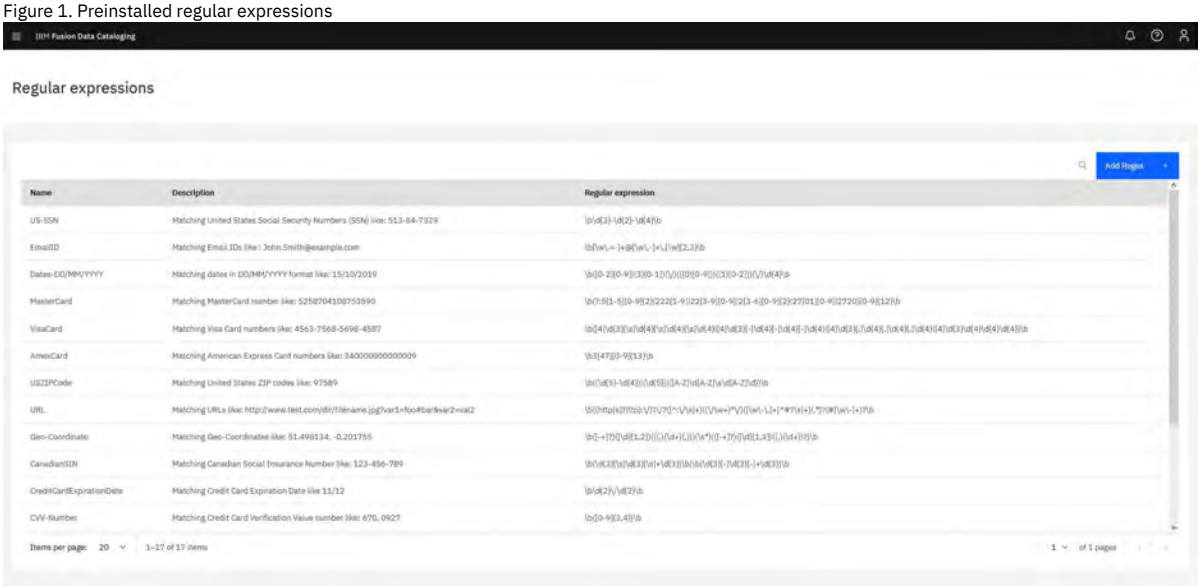
About this task

Validate that the regular expressions that are required for the policy are present. You can create or modify them if necessary.

Procedure

1. On the metadata page, select the Regular Expression tab.
   The list of available expressions is displayed.

   Figure 1. Preinstalled regular expressions



2. Search through the list of regular expressions to find any that match the content to be searched.
   As shown in Figure 1 that includes a selection of regular expressions.
   For example, an expression with an embedded value that might be extracted.

   ```
   ^([\w\.=-]+@[\w\.-]+\.[\w]{2,3})$
   ```

   This regular expression matches an email address, and the value that is returned for the tag is the matched email address. This sort of regex is appropriate to use in a value type match.
   Another example is an expression with no embedded value, for a straight match.

   ```
   ^Patient Name:.*$
   ```

   This expression detects the presence of the literal string **"Patient Name:"** and subsequent string in a fixed format file, but it does not extract the value. This is appropriate for use in a Boolean find search.

3. If there are no suitable regular expressions, select the Create icon.
4. If a regular expression exists but requires modifications select the expression and click the "Modify" icon. If this regular expression is in use by any other policy, this modification affects those policies.
5. Enter a suitable name, description, and the regular expression pattern.
6. Select Save.

Creating a content search policy

You can create a content search policy to filter a search query that finds candidate documents to which you can apply the policy.

About this task

For information about creating a content search policy, see Managing metadata policies and Adding policies.

Viewing content search application logs

The following information describes how to view the content search application logs.

About this task

If you want to view the content search application logs, perform the following actions.

Procedure

1. Run the following command for enter in the project:

   ```
   oc project ibm-data-cataloging
   ```

2. Any failures of download or inspection are logged in the application log file.
   The following command provides the application log file:

   ```
   oc logs deployment/isd-contentsearchagent
   ```

Hints and tips for using content search

Following are some best practices for using the content search feature.

Testing on a subset of documents

Running a content search policy on a set of documents has several steps that includes retrieving the document, formatting it as text, if necessary, and searching the document. Depending on the number, formatting, and size of the documents, searching the document can be a time-consuming process.

Therefore, it is best to test the policy and corresponding expressions on a small subset of documents to determine whether the policy and the regular expressions you select are correct. One way to run the test is to use a policy filter that selects only a small set of documents. After you confirm that the policy and search criteria is operating as expected, you can run it against the required set of documents.

The test on a subset of documents can also help you estimate how long the policy might take to run on the complete set of documents.

Avoiding retagging

When you rerun a policy against a set of documents that is previously tagged, the documents are retagged. If the values returned are different than the previous search, they are updated. This difference might occur if the policy or the set of expressions is modified, or if the set of documents is modified.

To avoid retagging the documents, add a criteria to the filter to not select documents that are already tagged.

Modifying regular expressions

If you modify a regular expression, it affects all policies that use that expression. Rerunning these policies might cause the documents to be tagged differently. To avoid changing the behavior of existing policies, create a new regular expression and use it in the specific policies where it is required.

Converting files with Apache Tika

Data cataloging uses Apache Tika to convert files to text before it searches the content. This conversion has an impact on the overall content search performance.

Therefore, files that are text format must be configured in the `contentsearch` agent to prevent them from being processed unnecessarily by Apache Tika. The default configuration treats JavaScript Object Notation (JSON) and Variant Call Format (VCF) file types as text. To add more text file types to the configuration, edit the file:

```
/opt/ibm/metaocean/data/agents/contentsearch/conf/contentsearch.conf
```

And add more types to the line:

```
text_filetypes=vcf,json
```

Apache Tika runs in a Kubernetes pod within Data cataloging. You can increase Apache Tika pod instances to improve performance. For example, run this command to scale the number of Tika instances to three:

```
oc -n ibm-data-cataloging scale --replicas=3 deploy/isd-tikaserver
```

Apache Tika is resource-intensive, so make sure that the number of Apache Tika instances does not exceed the host resources.

Supported connection types

Content search on Data cataloging supports the following connection types:

- Spectrum Scale
- COS
- NFS
- S3
- SMB

For more information, see the topic *IBM Spectrum® Scale data source connection* in *IBM Data Cataloging: Concepts, Planning, and Deployment Guide*.For more information, see IBM Storage Scale data source connections.

Tiering data by using ScaleILM application

Use the IBM Data Cataloging ScaleILM application to move data to different tiers (pools) that are configured on the IBM Storage Scale connection.

Before you begin

- Configure IBM Storage Scale with one or more internal pools. For more information, see: Internal storage poolsFor more information, see *Internal Storage pools* in the *IBM Storage Scale: Administration Guide*.
- If you want to, you can configure IBM Spectrum® Archive and its pools.
- Create the IBM Storage Scale connection in IBM Data Cataloging. If the external pool that is managed by IBM Storage Archive is used as the destination tier, then the "host" setting of IBM Storage Scale connection must specify one of the IBM Storage Archive nodes.
- IBM Data Cataloging ScaleILM application uses the same user, that is configured in IBM Data Cataloging to scan the IBM Storage Scale and run the Data Movement policies. For this data movement, the IBM Storage Scale connection is used as the 'source_connection'. For more information, see Creating or identifying a user ID and password for scanning. For more information, see the topic *Creating or identifying a user ID and password for scanning* in the *IBM Data Cataloging: Concepts, Planning, and Deployment Guide*.

About this task

The IBM Data Cataloging ScaleILM application provides the advanced data tiering function through the IBM Storage Scale connection. It uses the system metadata and custom metadata of documents that are collected by IBM Data Cataloging for this advanced data tiering function. This capability eliminates the need to scan file systems during IBM® License Manager (ILM) execution. It also provides cognitive tiering capability by using the results of IBM Data Cataloging's AUTOTAG, CONTENT SEARCH, and DEEP-INSPECT policies.

IBM Storage Scale provides the built-in Information Lifecycle Management (ILM) capability that optimizes the cost-effectiveness of data by moving the physical location of data between the storage pools with different cost or performance characteristics. IBM Storage Scale's ILM supports the data movement between its internal storage pools and between both the internal pool and the external storage repository. The movement from external storage repository is managed by external applications, such as IBM Storage Archive Enterprise Edition (EE) and IBM Storage Protect for Space Management.

Ensure that you know the IBM Storage Archive Enterprise Edition (EE) software versions that are supported. For more information, see IBM Spectrum Storage software requirements. For more information, see the topic *IBM Spectrum Storage software requirements* in the *IBM Data Cataloging: Concepts, Planning, and Deployment Guide*.

While tiering data in external pools, the current location of the data is indicated in its file state. The data locations and the corresponding file states that indicate these locations, are listed in the following table:

Table 1. Data locations and File States

Data locations	File states
The data location is only in the internal pool.	The file state is resident.
The data location is both internal and external pool.	The file state is premigrated.
The data location is only in the external pool.	The file state is migrated.

Note: The functions of the ScaleILM application depend on the capability of underlying storage hardware and its management software. For more information, see the following topics:

- *Information Lifecycle Management for IBM Storage Scale* in the *IBM Storage Scale: Administration Guide*
- *IBM Storage Archive Enterprise Edition (EE) IBM Documentation*

Procedure

1. Log in to IBM Data Cataloging GUI.
2. Go to Admin > Management Policies page.
3. Click Add Policy to create a policy.
4. Click the slider control and set the status to one of the following values:

   Active
   : An active policy is run whenever its scheduling event is reached.

   Inactive
   : An inactive policy is not run even when its scheduling event is reached, including the Now event.

5. Enter a policy name.
6. Enter a Policy filter. The policy filter includes the criteria for selecting the files for tiering, such as `filetype="pdf"`.
   Note: If the destination tier is the external pool that is managed by IBM Storage Archive, define the policy filter criteria as "state in 'resdnt'". For example, `filetype ='txt'` and `state in 'resdnt'`.
   If the filter criteria do not include `state in 'resdnt'` while tiering data to an external pool, the ScaleILM application skips the files that are not in resdnt state.
7. Click Next Step to select the type of policy.
8. Select `TIER` as the Policy type.

9. Select `ScaleILM` as the Agent.
10. Select the `source_connection` from the drop-down list. The source connection is the source from where the data is being moved, such as the name of the defined IBM Storage Scale connection.
11. Enter the `destination_tier` where you want to move your data, such as:
    - Internal Pools
      Gold, silver, bronze, flash system
    - External Pools
      - **`archive:pool1@library1`**
      - **`archive:pool2@library2`**
      - **`archive:pool1@library1, pool2@library3`**

    Note: When you specify an IBM Storage Scale internal pool as the "destination_tier", you need to ensure that you specify a valid internal pool name. That internal pool name must be configured in the IBM Storage Scale source connection for the corresponding data source (file system). These internal pools can be listed by using the following IBM Storage Scale command:

    ```
    mmlspool <device> all
    ```

    When you specify an external pool that is managed by IBM Storage Archive as the "destination_tier", you must specify a valid archive pool. This archive pool must be defined in the IBM Storage Archive that is configured on the IBM Storage Scale cluster node.

    Additionally, you must specify the name of the pools that are defined in the IBM Storage Archive configuration, with the prefix "archive:". The pool names must be specified in the same format as defined in the -p option of IBM Storage Archive Enterprise Edition (EE) CLI (eeadm). The syntax must be as follows:

    ```
    archive:<poolName>@<libraryName>
    ```

    or

    ```
    archive:<poolName1>@<libraryName1>,<poolName2>@<libraryName2>, ...
    ```

    The policy execution fails when you do not follow the instructions.

12. Select Next Step to enter a schedule. The schedule indicates when you want to start the tiering.
13. Select Next Step to review the policy.
14. Select Submit to create the policy.
15. When the policy is created, view the files on the IBM Data Cataloging Search catalog page to ensure that they are moved to the new tier. The following metadata are updated:

    Tier
    : Displays the name of the internal pool in IBM Storage Scale where the data is stored.
      Note: Even if the file is in the migrated state (migrtd), the tier field shows the name of the original internal pool.

    State
    : Displays the current state as one of the following values:
      - resdnt
      - premig
      - migrtd

    Migloc
    : Displays the location information of the external pool when the file is in premigrated (premig) or migrated (migrtd) state. If the file is in a resident (resdnt) state, this field shows `NA`.

    SizeConsumed
    : Displays the actual size of the file in bytes, that is associated with the IBM Storage Scale file system field SizeConsumed Bytes.

- **Viewing ScaleILM application logs**
  The following section describes how to view the ScaleILM Data Mover application logs.

Viewing ScaleILM application logs

The following section describes how to view the ScaleILM Data Mover application logs.

About this task

Follow the procedure to view the ScaleILM Data Mover application logs.

Procedure

1. Run the following command:

   ```
   oc -n ibm-data-cataloging logs deployment/isd-scaleilmdatamover
   ```

2. Any failures that occur while the files are processed for tiering by ScaleILM Data Mover application, are logged in the application log file. A sample error log from the ScaleILM Data Mover application logs is shown in the following section. These error logs occur when the file is not on the specified `'source_connection'` while the application is processing the Tiering policy.

   ```
   2020-04-28 14:44:36,439 - __main__ - ERROR - File not found:
   /ibm/scale0/unicode_test/migrateDir.popFSDir.19824/unicodedir/test010.\n.test
   2020-04-28 14:44:36,440 - __main__ - ERROR - File not found:
   /ibm/scale0/unicode_test/migrateDir.popFSDir.19824/unicodedir/test137..test
   2020-04-28 14:44:36,440 - __main__ - ERROR - File not found:
   /ibm/scale0/unicode_test/migrateDir.popFSDir.19824/unicodedir/test147..test
   ```

```
2020-04-28 14:44:36,440 - __main__ - ERROR - File not found:
/ibm/scale0/unicode_test/migrateDir.popFSDir.19824/unicodedir/test157..test
```

A sample error log from the ScaleILM Data Mover application logs for an invalid `'destination_tier'`, is shown in the following section:

```
2020-04-28 14:35:39,566 - __main__ - ERROR - Invalid destination_tier: 'gold:abcd'.
```

A sample error log from the ScaleILM Data Mover application logs when the `'destination_tier'` does not exist on the IBM Storage Scale connection, is shown in the following section:

```
2020-05-12 14:32:38,436 - __main__ - ERROR - Provided internal pool goldflower does not exist on system: 9.11.212.29
```

3. Run the following command to view policy engine pod logs:

```
oc -n ibm-data-cataloging logs deployment/isd-policyengine
```

A sample policy engine pod log is shown in the following section. This log shows information that is received from the ScaleILM Data Mover application. It describes errors that are encountered while a Tiering policy is processed and when a file is not found on the specified `'source_connection'`:

```
2020-04-28 14:44:37,455 - policy.policyapiservice - ERROR - [policy_id: scale-tier-unicode-modevvm19-pol]: Agent reported
'status: failed' for fkey: 'modevvm19.tuc.stglabs.ibm.comscale0121547', path:
'/ibm/scale0/unicode_test/migrateDir.popFSDir.19824/unicodedir/test237..test', reason: 'File not found:
/ibm/scale0/unicode_test/migrateDir.popFSDir.19824/unicodedir/test237..test', errno: 'ENOENT'
2020-04-28 14:44:37,455 - policy.policyapiservice - ERROR - [policy_id: scale-tier-unicode-modevvm19-pol]: Agent reported
'status: failed' for fkey: 'modevvm19.tuc.stglabs.ibm.comscale0121557', path:
'/ibm/scale0/unicode_test/migrateDir.popFSDir.19824/unicodedir/test247..test', reason: 'File not found:
/ibm/scale0/unicode_test/migrateDir.popFSDir.19824/unicodedir/test247..test', errno: 'ENOENT'
```

A sample error log from the Policy engine pod logs for an invalid `'destination_tier'`, is shown in the following section:

```
2020-04-28 14:35:40,064 - policy.policyapiservice - ERROR - [policy_id: scale-tier-modevvm19-invalid-destn-pol]: Agent
reported 'status: failed' for fkey:  'modevvm19.tuc.stglabs.ibm.comscale0300770', path: '/ibm/scale0/covid-19-
dataset/inference/NWPU-Chest-X-Ray/2b16346f-8fe4-45ee-b3c7-eaa46fd26041.png', reason: 'Invalid destination_tier:
'gold:abcd'.', errno: 'EINVAL'
```

A sample error log from the Policy engine pod logs when the `'destination_tier'` does not exist on the IBM Storage Scale connection, is shown in the following section:

```
2020-05-12 07:40:34,819 - policy.policyapiservice - ERROR - [policy_id: scale-tier-pol]: Agent reported 'status: failed'
for fkey: 'nikkogpfs26728', path: '/ibm/gpfs2/scale0/customer_e2e/financial_report_file6.pdf', reason: 'Provided internal
pool goldflower does not exist on system: 9.11.212.29', errno: 'EINVAL'
2020-05-12 07:40:34,819 - policy.policyapiservice - ERROR - [policy_id: scale-tier-pol]: Agent reported 'status: failed'
for fkey: 'nikkogpfs26748', path: '/ibm/gpfs2/scale0/customer_e2e/HR_report_US_file6.pdf', reason: 'Provided internal pool
goldflower does not exist on system: 9.11.212.29', errno: 'EINVAL'
```

Copying data by using ScaleAFM application

The ScaleAFM application supports copy function between IBM Storage Scale connection and IBM Cloud® Object Storage connection.

Before you begin

- You must know the IBM Storage Scale versions that are supported. For more information, see IBM Spectrum® Storage software requirements. For more information, see *IBM Spectrum Storage software requirements* in the *IBM Data Cataloging: Concepts, Planning, and Deployment Guide*.
- You need to configure the Active File Management (AFM) functions on IBM Storage Scale filesets under a filesystem. With this configuration, AFM now works as a cache for a bucket or vault on a IBM Cloud Object Storage, or other Cloud Object Storage data source that supports Amazon S3 protocol as a target.

  For more information, see the topic *Introduction to AFM to cloud object storage* in the IBM Storage Scale *Concepts, Planning, and Installation Guide*.
- You need to create the IBM Storage Scale connection in IBM Data Cataloging by configuring the filesystem in the IBM Storage Scale cluster, which contains the AFM fileset, with the Cloud Object Store target vault that is already configured.
- IBM Data Cataloging ScaleAFM application uses the same user that is configured in IBM Data Cataloging to scan the IBM Storage Scale connection and run the data movement policies. For this data movement, the IBM Storage Scale connection is used as the destination_connection . For more information, see Creating or identifying a user ID and password for scanning. For more information, see *Creating or identifying a user ID and password for scanning* in the *IBM Data Cataloging: Concepts, Planning, and Deployment Guide*.

About this task

The IBM Data Cataloging ScaleAFM application provides the advanced copy function between the IBM Storage Scale connection and the IBM Cloud Object Storage connection (IBM Cloud Object Store or an S3 connection).

The application uses the system metadata and custom metadata of the documents that are collected by IBM Data Cataloging. The IBM Data Cataloging collects the metadata for advanced data management function to copy the select set of data from a vault on the IBM Cloud Object Storage connection to an AFM fileset configured on a IBM Storage Scale connection.

The application also provides the cognitive data management capability by analyzing the results of IBM Data Cataloging's AUTOTAG, Content Search, and Deep Inspect policies.

The AFM to Cloud Object Storage is an IBM Storage Scale feature that enables placement of files or objects in an IBM Storage Scale cluster to a Cloud Object Storage. Cloud object services such as Amazon S3 and IBM Cloud Object Storage offers industry-leading scalability, data availability, security, and performance.

The AFM to Cloud Object Storage allows associating an IBM Storage Scale fileset with a Cloud Object Storage. You can use a Cloud Object Storage to store large amount of data for use cases: mobile applications, backup and restore, enterprise applications, and big data analytics. The data is stored locally on IBM Storage Scale filesets and on

a Cloud Object Storage. For more information, see the topic *Introduction to AFM to cloud object storage* in the IBM Storage Scale *Concepts, Planning, and Installation Guide*.

Note:
The functions of ScaleAFM application depend on the capability of the underlying storage hardware and its management software, specifically the IBM Storage Scale's AFM functions for Cloud Object Store.

For more information, see the topic *Active File Management for IBM Spectrum Scale* in the *IBM Spectrum Scale: Administration Guide*.

Procedure

1. Log in to IBM Data Cataloging GUI.
2. Go to Administration > Data Management page.
3. Click Add Policy to create a policy.
4. Enter a policy name.
5. Select `COPY` from the Type list.
6. Click Proceed to Configure.
7. Select `ScaleAFM` from the Application list.
8. Enter a Policy filter in the Filter field.
   The policy filter includes the criteria for selecting the files for copying from source Cloud Object Store datasource to a destination IBM Storage Scale datasource. For example, the policy filter can be like **`filetype="pdf"`**.
9. Select the connection type from the Source Connection list.
   It comprises connections that are configured with IBM Data Cataloging of type "Cloud Object Storage" or "Simple Storage Service (S3)". The source connection is the source from where the data is being copied, such as the name of the defined IBM Cloud Object Storage connection.
10. Select the connection type from the Destination Connection list.
    It comprises connections that are configured with IBM Data Cataloging of type 'Spectrum Scale'. The destination connection is the target where the data is being copied, such as the name of the defined IBM Spectrum Scale connection.
    Note: The IBM Storage Scale file system which is configured as the datasource for the destination IBM Storage Scale connection needs to be already configured with a fileset with IBM Storage Scale AFM functions enabled and configured with a Cloud Object Storage vault as the target. This vault must match with the one configured as `Source Connection` in step 9.
11. Enter the fileset value in Spectrum scale_afm_fileset field.
    It is configured as the AFM fileset under the destination IBM Storage Scale connection, as explained in step 10.
12. Enter the directory path in the Target base dir field.
    The Target Base Dir defines the absolute path of a base directory on the destination IBM Storage Scale system to which files from the source system are to be downloaded. It must be a directory path that is expressed as absolute path under the link path of the destination IBM Storage Scale fileset. The field name must begin with a '/'. If not specified, the link path of the IBM Storage Scale AFM fileset that is configured with IBM Storage Scale connection, is used as the **`target_base_dir`**.
13. Click Proceed to Schedule to configure a schedule to the policy.
14. Click the slider control to select one of the following policy status:

    Active
    : An active policy is run whenever its scheduling event is reached.

    Inactive
    : An inactive policy is not run even when its scheduling event is reached, including the Now event.

15. Select a policy schedule under Frequency.
    The schedule indicates when you want to schedule the policy for an execution.
16. Click Proceed to review to review the policy.
17. Click Submit to create the policy.
    - When the policy is completed, you can update the IBM Data Cataloging catalog for the destination IBM Storage Scale connection by manually starting a scan of the connection.
    - If the destination IBM Storage Scale connection is already created enabling live events, then the IBM Data Cataloging catalog automatically gets updated in a while with the entries relevant to the newly copied files from the Cloud Object Storage datasource by using the ScaleAFM data management policy, recently executed.

- **Viewing ScaleAFM application logs**
  Procedure to view ScaleAFM application logs.

Viewing ScaleAFM application logs

Procedure to view ScaleAFM application logs.

Procedure

1. Issue the following command to view the ScaleAFM application logs:

   ```
   oc -n ibm-data-cataloging logs deployment/isd-scaleafmdatamover
   ```

2. Any failures that occur while the files are processed for copying by ScaleAFM Data Mover application, are logged in the application log file. A sample error log from the ScaleAFM Data Mover application when a file (object) does not exist in the vault. The vault is configured with the source Cloud Object Storage data source, while the application is processing the Copy policy.

   ```
   2020-09-30 15:07:47,032 - __main__ - ERROR - Failed queuing the file 'vault1/1ws3711.txt' during AFM data transfer
   ```

   This error log occurs when the IBM Storage Scale AFM Fileset name entered in the policy definition, does not exist on the IBM Storage Scale filesystem that is configured as datasource in the destination IBM Storage Scale connection:

   ```
   2020-09-30 15:02:12,549 - __main__ - ERROR - An exception occurred while processing message: Fileset 'DAAA_ICOS123' not found in file system 'fs1'
   ```

3. Run the following command to view policy engine pod logs:

   ```
   oc -n ibm-data-cataloging logs deployment/isd-policyengine
   ```

   A sample policy engine pod log error from the ScaleAFM Data Mover application when a Copying policy is processed and when a file is not found on the specified `'source_connection'`:

   ```
   2020-09-30 15:07:47,080 - policy.policyapiservice - ERROR - [policy_id: DAAA_Pol1]: Agent reported 'status: failed' for
   fkey: 'vault11ws3711.txt'
   , path: 'vault1/1ws3711.txt', reason: 'Failed queuing the file 'vault1/1ws3711.txt' during AFM data transfer', errno:
   'ENOENT'
   ```

   A sample error log from the policy engine pod logs, when the IBM Storage Scale AFM Fileset name entered in the policy definition, does not exist on the IBM Storage Scale filesystem. The IBM Storage Scale filesystem is configured as datasource in the destination IBM Storage Scale connection:

   ```
   2020-09-30 15:02:13,100 - policy.policyapiservice - ERROR - [policy_id: DAAA_InvFileset]: Agent reported 'status: failed'
   for fkey: 'vault1analyt
   ics_results1.txt', path: 'vault1/analytics_results1.txt', reason: 'An exception occurred while processing message: Fileset
   'DAAA_ICOS123' not fou
   nd in file system 'fs1'', errno: 'EIO'
   ```

Exporting metadata to IBM Watson® Knowledge Catalog

IBM Data Cataloging Watson Knowledge Catalog (WKC) connector supports export of the metadata to either an IBM Cloud® instance or to an On-Premise Instance of the Watson Knowledge Catalog .

The default deployment of the WKC connector contains the parameters that are associated with the IBM Cloud instance of Watson Knowledge Catalog . On start, the IBM Data Cataloging WKC connector waits for WKC credential values. In the absence of appropriate credentials, the Export button is not visible on the IBM Data Cataloging user interface. For more information, see Configuring Watson Knowledge Catalog connector.

- **Configuring Watson Knowledge Catalog connector**
  A utility script is provided to simplify and automate the Watson Knowledge Catalog connector configuration. Alternatively, you can configure the Watson Knowledge Catalog connector manually.
- **Exporting metadata**
  Export data with relevant metadata tags from IBM Data Cataloging to Watson Knowledge Catalog .
- **Mapping similar source connections in Watson Knowledge Catalog**
  IBM Data Cataloging supports mapping source connections with Watson Knowledge Catalog connections (WKC) through WKC connector App.
- **Troubleshooting export issues**
  This topic describes some issues faced while exporting data by using the Watson Knowledge Catalog.

Configuring Watson Knowledge Catalog connector

A utility script is provided to simplify and automate the Watson Knowledge Catalog connector configuration. Alternatively, you can configure the Watson Knowledge Catalog connector manually.

Before you begin

Collect details of the IBM Cloud® Watson Catalog instance. The details that are required can vary depending on whether the IBM Watson® Catalog is on Cloud or On-premise.
To add IBM Cloud Watson Catalog details to the IBM Data Cataloging, obtain the following details:

1. Obtain the API key for accessing the Watson Knowledge Catalog application. To access WKC application on the IBM® Cloud, Log in to IBM Cloud and select Create an IBM Cloud API key.
2. Export or copy the key.
3. Copy the base URI of your WKC instance. For IBM Cloud, the WKC instance URI is based on the geographical location.

To add the IBM on-premises Watson Knowledge Catalog details to IBM Data Cataloging, obtain the following details:

- Obtain the values for the following Watson Knowledge Catalog (WKC) parameters from the systems administrator of the On-Premises instance:
  - WKC_USER
  - WKC_PASSWORD
  - WKC_BASE_URI
  - WKC_AUTH_URI

To successfully export document-related metadata from IBM Data Cataloging to the Watson Knowledge Catalog , WKC application must have access to the original documents. This step is achieved by using either a linked or a non-linked data source. Exporting metadata from a linked data source indicates that both Watson Knowledge Catalog and IBM Data Cataloging have access to the original data source location, eliminating the need to copy the original files.

Exporting metadata from a non-linked data source is used when Watson Knowledge Catalog does not have access to original IBM Data Cataloging data source location. In this case, IBM Data Cataloging copies the files that are associated with the export metadata to a location where the Watson Knowledge Catalog service can access these documents. A Watson Knowledge Catalog Cloud Object Storage (COS) connection must be available to export from non-linked data sources.

You must know the details of linked or non-linked data sources before you start configuring the WKC Connector. For more information, see Mapping similar source connections in Watson Knowledge Catalog.

- **Configuring Watson Knowledge Catalog using the utility script**
  Utility script is used to configure the Watson Knowledge Catalog (WKC) and the script automates the WKC connector app configuration process.

- **Configuring Watson Knowledge Catalog manually**
  Procedure to configure Watson Knowledge Catalog (WKC) connector app manually.

Configuring Watson Knowledge Catalog using the utility script

Utility script is used to configure the Watson Knowledge Catalog (WKC) and the script automates the WKC connector app configuration process.

Procedure

Log in to the IBM Data Cataloging instance and run the following command:

```
/opt/ibm/metaocean/utility-scripts/configureWKC.sh
```

The script prompts for credentials and data source connections and configures the WKC application.

The script can be rerun to change the configuration. This is applicable for the following scenarios:

- Exporting from IBM Data Cataloging to a new or different Watson Knowledge Catalog connection.
- A new connection is added to IBM Data Cataloging.

Configuring Watson Knowledge Catalog manually

Procedure to configure Watson Knowledge Catalog (WKC) connector app manually.

Before you begin

Before you perform any changes to the Watson Knowledge Catalog (WKC) connector application configuration, ensure that no WKC export policies are in running state. If you do any changes in the WKC connector application configuration, while policies are running it can cause the WKC connector application to fail to start and resulting in a **`CrashLoopBackoff`** state. If you face this issue, stop any WKC export policies that are running and the WKC connector application starts normally.

Procedure

1. Log in to the IBM Data Cataloging instance and issue the following command to configure the WKC connector app:

   ```
   oc -n ibm-data-cataloging edit deploy/isd-wkcconnector
   ```

2. For IBM® Watson Knowledge Catalog cloud deployment, define the following parameters:
   - name:WKC_API_KEY
   - value:<API-KEY-VALUE>

   For cloud deployment, ensure that **`WKC_USER`** and **`WKC_PASSWORD`** value remain undefined in the environment. To delete WKC connector secret, issue the following command:

   ```
   oc delete secret wkcconnector --ignore-not-found
   ```

   You can set WKC_BASE_URI value for WKC cloud, depending on the on-premises geographical location. The parameter value is the default when WKC_BASE_URI is not defined.

   Example

   ```
   name: WKC_BASE_URI
   value:https://api.dataplatform.cloud.ibm.com/v2/
   ```

3. For IBM Watson Knowledge Catalog on deployment, add the parameter details in the following sample format:

   ```
   name:WKC_BASE_URI
   value:https://<wkc_hostname>/v2/
   name:WKC_AUTH_URI
   value:https:// <wkc_hostname>/icp4d-api/v1/authorize
   ```

   a. To define the WKC connector secret manually, issue the following command:

   ```
   oc create secret generic wkcconnector --from-literal="user=$WKC_USER" --from-literal="password=$WKC_PASSWORD"
   ```

4. Configure any linked data sources if needed by setting the value of the `WKC_CONNECTION_MAP` field.
   For more information see, Mapping similar source connections in Watson Knowledge Catalog.Mapping similar source connections in Watson Knowledge Catalog on page 40.
   Note:
   - If you switch to a different IBM Watson Knowledge Catalog , you can edit the relevant WKC environment variables in the deployment editor to add the values that are associated with the new account. The application automatically restarts and recognizes the new account that is linked.
   - If you add new catalogs to the WKC instance, IBM Data Cataloging retains the old registration information and continues to point to the old catalog IDs. To resolve this issue, follow the procedure:
     - Go to metadata > Applications.
     - To restart the WKC application instance, issue the following commands:

       ```
       oc -n ibm-data-cataloging delete isd-wkcconnector
       ```

Exporting metadata

Export data with relevant metadata tags from IBM Data Cataloging to Watson Knowledge Catalog .

About this task

The Watson Knowledge Catalog is a data catalog system that is not always able to scan relevant data sources and capture the relevant metadata from those files. IBM Data Cataloging helps to bridge this critical gap by helping to export data to Watson Knowledge Catalog with all relevant metadata tags.

Procedure

1. On the IBM Data Cataloging Dashboard, search for the data to be exported by using a specific filter criteria.
2. Click Export Data. The Export Data to Watson Knowledge Catalog window appears.
3. Under Destination Catalog, select the catalog in Watson Knowledge Catalog where you want to export the data.
4. Select the tags that you want to export from the list in Metadata Tags to Export.
5. Click Submit.
6. After completion of the process, the exported data is displayed in the Watson Knowledge Catalog with the tags that are imported from IBM Data Cataloging.
   Now you have an alternative option to create a data management policy to export data to the Watson Knowledge Catalog by using the DATASET_EXPORT policy type.
   Note: Tags in IBM Data Cataloging represent a name (for example, SizeRange) and a value (for example, small, large, or medium). In Watson Knowledge Catalog , the tags represent a value. The exported data maps both of these attributes and it creates a single label.
   For example, `SizeRange:Small`.

Mapping similar source connections in Watson Knowledge Catalog

IBM Data Cataloging supports mapping source connections with Watson Knowledge Catalog connections (WKC) through WKC connector App.

When metadata is being exported to WKC, IBM Data Cataloging might use a source connection that WKC can also connect to. In such a scenario, you can configure the connection mapping within the WKC Connector App.

The connections that can be linked to are:

- Amazon S3
- IBM Cloud® Object Storage
- IBM Storage Scale

Note: Connections with IBM Storage Scale are established through an S3 connection as WKC does not support IBM Storage Scale directly.
The details for configuring the connection maps with each of these source connections are described.

S3

For an IBM Data Cataloging S3 connection, the WKC connection must contain the following details:

Bucket
: If the bucket name is configured, then you do not need to provide any further configuration details. The WKC Connector App can infer the details from the global namespace of Amazon S3 buckets.
  If the bucket name is not provided, then configure the *WKC_Connection_Map* environment variable by using the following format: `<datasource>;<cluster>:<wkc connection name>`. A sample variable value is shown. All documents, that the WKC App receives in its work message corresponding to the data source and cluster pair that is defined in the variable, maps to the WKC connection of that name.

  ```
  WKC_CONNECTION_MAP=testbucket1.sd.ibm.com;s3.eu-west-1.amazonaws.com:s3_con_no_bucket
  ```

  Note: It is mandatory to provide the bucket name while you are configuring details in IBM Data Cataloging but it is optional for WKC.

IBM Cloud Object Storage Infrastructure

For an IBM Data Cataloging IBM Cloud Object Storage connection the WKC connection must contain the following details:

Login URL
: The login URL must be that of the accessor defined in IBM Data Cataloging. The data source for IBM Cloud Object Storage is the vault. However, since it is not possible to provide a vault here, you must provide the mapping within the environment variable *WKC_CONNECTION_MAP* in the following format: `<datasource>;<cluster>:<wkc connection name>`. A sample variable value is shown.

  ```
  WKC_CONNECTION_MAP=vault1;e09cdac0-80f8-73be-00ed-cb8edeede242:local_cos_con
  ```

  Multiple IBM Cloud Object Storage connections to the same system can map to the same WKC connection (as it is at a higher level and can see all vaults).

IBM Storage Scale

Connection mapping with IBM Storage Scale must be done through an S3 connection as WKC cannot connect directly with it.

To establish an S3 connection, configure the following details in the WKC connector app:

Endpoint URL
: The S3 Endpoint URL on the IBM Storage Scale mode. For example, `http://modevvm19.tuc.stglabs.ibm.com:9000`.

Access Key
: Type the S3 access key.

Secret Key
: Type the S3 secret key.

Bucket
: Do not enter values in the bucket field.

Define the mapping within the environment variable *WKC_CONNECTION_MAP* in the following format: `<datasource>;<cluster>:<wkc connection name>`. A sample variable value is shown:

```
WKC_CONNECTION_MAP=scale0;modevvm19.tuc.stglabs.ibm.com:s3_scale
```

Mapping multiple connections

You can map multiple connections within the same environment variable by using commas to separate the values. A sample is shown:

```
WKC_CONNECTION_MAP=vault1;e09cdac0-80f8-73be-00ed-
cb8edeede242:local_cos_con,scale0;modevvm19.tuc.stglabs.ibm.com:s3_scale,testbucket1.sd.ibm.com;s3.eu-west-
1.amazonaws.com:s3_con_no_bucket
```

Run the following command to edit the WKC Connector app deployment and add the mapping:

oc -n spectrum-discover edit deploy/spectrum-discover-wkcconnector

The WKC connections are configured in the following format:

```
name: WKC_CONNECTION_MAP
value: vault1;e09cdac0-80f8-73be-00ed-
cb8edeede242:local_cos_con,scale0;modevvm19.tuc.stglabs.ibm.com:s3_scale,testbucket1.sd.ibm.com;s3.eu-west-
1.amazonaws.com:s3_con_no_bucket
```

Note: The **edit** command configures the WKC connection mapping and automatically restarts the WKC pod.

Troubleshooting export issues

This topic describes some issues faced while exporting data by using the Watson Knowledge Catalog.

- **Authentication failure with Watson Knowledge Catalog - both on-Premise and IBM Cloud**
  Resolve authentication failure with the Watson Knowledge Catalog (WKC).
- **Incorrect WKC URL configuration**
  Resolve the errors that occur due to an incorrect WKC URL configuration.
- **Invalid connection type in catalog**
  Resolve the error that occurs due to an invalid connection type in catalog.
- **No linked connection type**
  Resolve the errors that occur when links cannot be identified for the connection type.
- **WKC connector pod in CrashLoopBackoff state**
  Check the following settings if the Watson Connector pod is in the CrashLoopBackoff or Error state.
- **S3 connection issues**
  Resolve the issues faced owing to S3 connection.

Authentication failure with Watson Knowledge Catalog - both on-Premise and IBM Cloud

Resolve authentication failure with the Watson Knowledge Catalog (WKC).

The following error is displayed on the Watson Knowledge Catalog connector logs, when the connector fails to authenticate with the Watson Knowledge Catalog instance.

```
2020-07-16 12:46:04,870 - WKCConnector - ERROR - User authentication failed with response code 401
2020-07-16 12:46:04,871 - WKCConnector - INFO - No valid token available, cannot connect to WKC. Please set API Key or User
Credentials
```

When the WKC authentication fails, check to ensure that the correct values are defined for the following environment variables for the on-premises connection:

```
WKC_USER, WKC_PASSWORD
```

If you are using an IBM Cloud® connection type, you must check the value of the environment variable `WKC_API_KEY`.

Run the following command to check and confirm the configuration for the specified environment variables in the pod helm chart: **oc edit deployment spectrum-discover-wkcconnector**

Incorrect WKC URL configuration

Resolve the errors that occur due to an incorrect WKC URL configuration.

The following errors occur:

```
HTTPSConnectionPool(host='machine.ibm.com', port=443): Max retries exceeded with url: /v2/catalogs (Caused by
NewConnectionError('<urllib3.connection.HTTPSConnection object at 0x7f63e7973c18>: Failed to establish a new connection: [Errno
-2] Name or service not known',))
2020-07-09 22:59:29,412 - WKCConnector - ERROR - Cannot communicate with WKC to retrieve catalog information. Check config
URIs.
```

A possible reason for the error is that invalid values are defined for the environment variable **WKC_BASE_URI**. You can get similar errors if you define incorrect values for the **WKC_AUTH_URI** environment variable.

Run the following command to check and confirm the configuration of the said environment variables in the pod helm chart: **oc edit deployment spectrum-discover-wkcconnector**.

The following error occurs, when incorrect API key is provided for WKC cloud connections:

```
2020-09-28 13:29:30,189 - WKCConnector - ERROR - Supplied API Key failed with response code 400
2020-09-28 13:29:30,190 - WKCConnector - ERROR - Unable to connect to WKC, resetting agent registration
```

The following error occurs, when incorrect user/password is provided for WKC on-premises connections:

```
2020-09-28 13:34:24,530 - WKCConnector - INFO - API Key not set, checking for user/password
2020-09-28 13:34:24,767 - WKCConnector - ERROR - User authentication failed with response code 401
2020-09-28 13:34:24,768 - WKCConnector - ERROR - Unable to connect to WKC, resetting agent registration
```

Invalid connection type in catalog

Resolve the error that occurs due to an invalid connection type in catalog.

When a valid connection type cannot be detected, the following error occurs:

```
2020-06-30 22:37:09,628 - WKCConnector - ERROR - Could not find a supported connection in the supplied catalog
```

This error might occur in the scenario when the connection type within the Watson Knowledge catalog (WKC) does not match the connection type on Spectrum Discover. For example, if you try to export metadata from an IBM Data Cataloging Cloud Object Storage (COS) source to a WKC catalog, which does not have an IBM Cloud® Object Storage connection. The error can also occur if you have an IBM Cloud Object Storage bucket on WKC mapped to an on-Premises IBM Cloud Object Storage bucket that is configured on IBM Data Cataloging.

No linked connection type

Resolve the errors that occur when links cannot be identified for the connection type.

The following errors can occur resulting in an export failure:

```
2020-07-01 13:58:34,129 - WKCConnector - ERROR - Cannot export /Changing ACEs/Not for Luke and Amy/Full Gamora/Partial
Luke/Partial Amy/No Clara/I mean not clara/TinyDoc.txt as document is not on a linked connection, and was unable to be
downloaded from the SD connection for copy
2020-07-01 13:58:34,129 - WKCConnector - ERROR - Failed to export /Changing ACEs/Not for Luke and Amy/Full Gamora/Partial
Luke/Partial Amy/No Clara/I mean not clara/TinyDoc.txt
```

An exported document must either be on a linked connection, or must be available for copying to backup IBM Cloud® Object Storage in the WKC catalog, if that exists.

WKC connector pod in CrashLoopBackoff state

Check the following settings if the Watson Connector pod is in the CrashLoopBackoff or Error state.

Check that there are no Watson Knowledge Catalog export policies that are in running state when you perform any changes to Watson Knowledge Catalog connector application configuration.

Check the settings for the following environment variables if you are using IBM Cloud® Watson Knowledge Catalog:

```
WKC_API_KEY
```

Check the correct value for the following parameter if you are using IBM Cloud Watson Knowledge Catalog:

```
WKC_BASE_URI
```

Check the settings for the following environment variables if you are using an on-premises Watson Knowledge Catalog :

```
WKC_BASE_URI
WKC_AUTH_URI
WKC_CONNECTION_MAP
```

After successfully exporting records, if you change an environment variable to an invalid value and then export records again, you can end up with unprocessed messages in the Kafka buffer. You must clear these messages before you can change the invalid environment variable to the correct value. Run the following commands to clear the Kafka queue and then restart the Watson Knowledge Catalog connector application.

Clear Queue

/opt/kafka/bin/kafka-topics.sh --delete --zookeeper localhost:2181 --topic WKCConnector_work

Recycle Pod

oc -n ibm-data-cataloging delete pod -l role=wkcconnector

If the Watson Knowledge Catalog connector pod remains down after you have performed the preceding steps, do the following:

1. If the Watson Knowledge Catalog connector (WKC connector) pod in IBM Data Cataloging cannot be started, delete the existing pod using the following command:

   ```
   oc -n ibm-data-cataloging delete pod -l role=wkcconnector
   ```

2. Retrieve the logs from the WKC connector pod using the following command:

   ```
   oc -n ibm-data-cataloging logs deployment/isd-wkcconnector
   ```

3. Check for errors in the logs by running the following command:

   ```
   oc -n ibm-data-cataloging logs deployment/isd-wkcconnector | grep error
   ```

4. If the log terminates with an exception, check the policy engine log for errors by using the following command:

   ```
   pollog | grep error
   ```

5. Check if the following error message appears at the time of the exception:

   ```
   <time_stamp> - policy.policyapiservice - ERROR - b'Error: Cannot update application, 'WKCConnector, it is in use by the follow
   ```

6. The WKC connector pod might not start until the error shown in step 5 is resolved.
   Go to the Tag Management Policies section in the user interface and stop all running policies.

7. After the policies are stopped, then restart the WKC connector pod by deleting it:

   ```
   oc -n ibm-data-cataloging delete pod -l role=wkcconnector
   ```

S3 connection issues

Resolve the issues faced owing to S3 connection.

To enable export of IBM Storage Scale file metadata without copying the actual file to Watson Knowledge Catalog, the latter must be connected to IBM Storage Scale through an S3 interface.

If you are experiencing issues with IBM Storage Scale file exports, check whether the connection is present in the connector connection map and that the details are made available in Watson Knowledge Catalog. You must check to see that correct values are defined for the IP, port, access, and secret key. To preview the documents in the catalog, you must check that the S3 interface is available on the IBM Storage Scale system.

Importing externally curated tags for COS/S3 using import tags application

The import tags application is used to import a set of externally curated tags for Cloud Object Storage and S3 services.

Before you begin

The S3/COS (Cloud Object Storage) data source that is associated with the objects in the external tags file must be scanned before you run an import tags policy.

Tag names must be defined in IBM Data Cataloging before the Import Tags policy is run. You must refrain from using Restricted tag type and use Open or Characteristics tag types.

Note:

- You can create limited number of Open type tags and the tags must correspond to values in the header row of the comma-separated values (CSV) file. Tags that are not defined before you trigger the policy are not imported.
- The CSV file must be in the bucket that is defined in the data source.
- Only a single tag file is supported per policy.

Requirements of the external CSV file are listed as shown.

- The tags file must be in CSV format.
- The first row in the file must be a header row or line.
- The first column must be the full object path or name. For example, if the bucket is auto_data, and the object name is car1/image1.png, then the first column entry is auto_data/car1/image1.png.
- The value in the header row for the first column is not restricted by IBM Data Cataloging.
- The subsequent columns in the CSV file represent the tag values that can be imported into IBM Data Cataloging for the associated object records.
- The second through Nth entries in the header row must correspond to valid tags in IBM Data Cataloging that are defined before you run the import tags policy.
- Each entry in the CSV file must represent a unique file in the data source.

```
Example contents of a CSV file:
objectname,bus,tree,stop_sign,red_light,yellow_light,green_light,pedestrian
auto_data/car1/image1.png,1,3,0,1,1,0,1
auto_data/car2/image1.png,1,6,0,0,0,0,12
auto_data/car2/image2.png,1,3,0,2,1,0,1
auto_data/car3/image1.png,1,3,0,2,1,0,2
```

The following tags can be defined in IBM Data Cataloging from the records available in the CSV file:

- bus
- tree

- stop_sign
- red_light
- yellow_light
- green_light
- pedestrian

Note: Only a single tag import policy can be run at a time.

About this task

The IBM Data Cataloging import tags application allows a user with DATA ADMIN role to apply a pre-curated set of labels (tags) that are available in an external CSV file to S3/COS object records stored in IBM Data Cataloging.

For example, an external analytics job might generate tag information for a set of S3/COS objects, and save this information into a CSV file. The CSV file comprises an entry for each object that contains an object name and an associated list of labels or tags.

The import tags application can merge these tags into the associated object records in IBM Data Cataloging, extending the records with new information.

Procedure

1. Configure a COS/S3 data source connection for IBM Data Cataloging, with the vault or bucket that gets scanned later.
   The resulting system metadata that is indexed in IBM Data Cataloging is then enriched with the imported tag data.

   Example
   COS/S3 datasource connection name is **`camera_vault`** and the vault or bucket name is **`camera_lidar_semantic`**. Connection is created and scan is run.

   For more information about configuring and scanning COS/S3 data sources, see Configure data source connections. For more information about configuring and scanning COS/S3 data sources, see *Configure data source connections* in the *IBM Data Cataloging: Concepts, Planning, and Deployment Guide*
2. Place external tag file on to the data source so that IBM Data Cataloging can access it.
   A single CSV file comprises a list of objects and tags. The objects and tags are applied to the indexed records for a COS/S3 data source. These objects and tags must be uploaded to the configured source storage bucket by using any IBM® COS/S3 compatible data management utility.

   Example
   IBM Data Cataloging data source with connection name **`camera_vault`** is configured to scan vault and bucket **`camera_lidar_semantic`**.

3. Define tag names in IBM Data Cataloging.
   a. Use the headers from the CSV file, starting with the second column.
   b. Create corresponding tags in IBM Data Cataloging for any of the columns where you want to import data.

   For example, for a column you want to import with header value ColumnA, create a tag ColumnA. The columns for which you do not want to import data, must not store tags that are defined with the header value.

   Example
   CSV file comprises columns with header values front, rear, center, car, bicycle, pedestrian, and truck that you want to import. Characteristics tags are created with names: front, rear, center, car, bicycle, pedestrian, and truck.

   For more information about creating tags, see Creating tags.
   For more information about creating multiple tags through REST API see, /policyengine/v1/tags/:POST.For more information about creating multiple tags through REST API, see */policyengine/v1/tags/:POST* in the *IBM Data Cataloging: REST API Guide*.
4. Initiate the policy by the using REST API interface. For more information, see Initiating Policy Using REST API.

- **Initiating Policy Using REST API**
  Initiating a policy using the REST API triggers the import tag policy engine to process metadata tags for the configured data sources.

Initiating Policy Using REST API

Initiating a policy using the REST API triggers the import tag policy engine to process metadata tags for the configured data sources.

- **Registering import tags application**
  The process of registering an import tags application.
- **Defining the import tag policy type**
  Procedure to define the import tags policy in a JSON payload.
- **Viewing the policy status**
  You can view policy status either from the IBM Data Cataloging GUI or by running read (GET) request.
- **Viewing the import tags application log**
  You can view the import tag application logs through a simple procedure.

Registering import tags application

The process of registering an import tags application.

About this task

The import tags application is built-in and pre-registered in IBM Data Cataloging. To view the registered application, you can use the IBM Data Cataloging GUI.

For more information, see Managing applications. Managing applications on page 63.

Defining the import tag policy type

Procedure to define the import tags policy in a JSON payload.

About this task

You can create the import tags policies by using the policy management REST API. A sample policy that is defined in a JSON payload is shown:

```
{
        "pol _id": "importtags_pol",
        "action_id": "IMPORT_TAGS",
        "action_params": {
                "agent":"Import Tags",
                "source_connection":"camera_vaul t",
                "tag_file_path":"camera_lidar _semantic/A2D2_labels.csv",
                "tag_file_type":"csv"
        },
        "schedule": "NOW",
        "pol_state": "active",
        "pol_filter": "datasource=' camera_lidar_semanti c' "
}
```

The following table lists the action parameters for defining the import tags policy type:

Table 1. List of action parameter for an import tags policy type

Action Parameter	Description
souce_connection	The Cloud Object Storage (COS) or S3 connection name that is defined in IBM Data Cataloging for the records that the imported tags can be applied to.
tag_file_path	The absolute path (bucket or object name) of the CSV file that contains the list of objects and associated tags to import.

IMPORT_TAGS is the new action_id that is defined for the import tags policies.

Procedure

1. Send the following query request to list all the supported action IDs on IBM Data Cataloging node:

   ```
   curl -H "Authorization:Bearer<token>"-k https://<data_cataloging_host>:443/policyengine/ v1/action_ids
   ```

2. Send the following query request to the IBM Data Cataloging node to obtain the current Import Tags application schema:

   ```
   curl-H" Authorization:Bearer <token>"-k https://<data_cataloging_host>:443/policyengine/v1/applications/ImportTags/schema?
   action_id=I MPORT_TAGS
   ```

3. To define the policy, and schedule or run it immediately, use the /policyengine/v1/policies -d '<data>': POST endpoint to process the request.
   For more information, see /policyengine/v1/policies -d '<data>': POST
   For more information, see */policyengine/v1/policies -d '<data>': POST* in the *IBM Data Cataloging: REST API Guide*.

Example

Send the following request to create an Import Tags policy for data source connection camera_vault that runs immediately by using the external tag file A2D2_labels.csv:

```
curl - k - H ' Authorization: Bearer <token>'
https: //<data_cataloging_host>/policyengine/v1/policies-d'
        {
                "pol_id": "importtags",
                "action_id": "IMPORT_TAGS",
                "action_params": {
                        "agent": "Import Tags",
                        "source_connection": "camera_vault",
                        "tag_file_path": "camera_lidar_semantic/A2D2_labels.csv",
                        "tag_file_type": "csv"
                },
                "schedule": "NOW",
                "pol_state": "active",
                "pol _filter": "datasource='camera_lidar _semantic' "
        }
'-XPOST -H" Content-Type:application/json"
```

The following response is returned:

```
Policy 'importtags_ut'added
```

Viewing the policy status

You can view policy status either from the IBM Data Cataloging GUI or by running read (GET) request.

About this task

Follow the procedure shown to view the policy status:

Procedure

1. Log in to IBM Data Cataloging GUI.
2. Go to Administration > Data Management tab.
   The policy status can also be checked with the following REST API read request:

   ```
   /policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET
   ```

   For more information, see /policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET.For more information, see */policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET* in the *IBM Data Cataloging: REST API Guide*.
   Note: As import tags policies rely on an external list of objects/file and not on a policy filter, the policy preview service does not apply to this policy type. The number of records updated in IBM Data Cataloging will be no more than the number of entries in the external tag file.
3. Go to Policies tab.
   Policies are listed with their status as active or inactive.

Example

A sample REST API request for checking import tag policy and the corresponding response is shown:

```
curl -k -H' Authorizati on:Bearer <token>'
https://<data_cataloging_host>/policyengine/v1/policies/importtags_ut

Response:
{
        "pol_id": "importtags_ut",
        "action_id": "IMPORT_TAGS",
        "action_params":"{\"agent\":\"Import Tags\", \"source_connection\": \"camera_vault\", \"tag_file_path\":
\"camera_lidar_semantic/A2D2_labels.csv\",\"tag_file_type\":\"csv\"}",
        "schedule":"NOW",
        "pol_filter":"datasource='camera_lidar_semantic' ",
        "pol_state":"active",
        "pol_status":"running",
        "explicit": "true",
        "execution_info":"{\"start_time\":\"2020-09-17  21: 04: 35\",\"total _count\":350,\"submitted_count\":
400,\"failed_count\": 8,\"completed_count\":300,\"ski pped_count\":0,\"i mport _tags_count\": 0,\"run_id\":\"a3b0f
7647647443e8c5237aebe2c5408\"}",
        "policy_owner": "sdadmin",
        "last_updated_by": "sdadmin",
        "last_updated":"2020-09-17T21:04:35.000Z",
        "collections":[]
}
```

During processing, the import tags application does some basic checking on the CSV file. If a row has fewer columns than the header row, the row is skipped. If a header value contains an invalid or missing tag name, the values for that column are not imported. These records are added to the failed count in the policy.

Viewing the import tags application log

You can view the import tag application logs through a simple procedure.

About this task

Follow the procedure to view the import tag application logs:

Procedure

Issue the following command to view the import tag application logs:

```
oc -n ibm-data-cataloging logs deployment/isd-importtags
```

A sample error log from the import tag application is shown in the following section. This error occurs when there exist rows in the CSV file that contain fewer columns than the header row:

```
2020-09-17 21:04:41,538-Import Tags-ERROR-Skipped  entry. Ragged row: expected 44 columns, found 36
```

A sample error log from the import tag application is shown in the following section. This error occurs when there exist empty rows in the CSV file (newline):

```
2020-09-17 21:04:41,540-Import Tags-ERROR-Skipped entry. Empty row.
```

A sample error log from the import tag application is shown in the following section. This error occurs when there exist rows with an invalid file/object name in the first column:

```
2020-09-17 21:04:56,239-Import Tags-ERROR-Skipped entry. Object/file name not found in csv column 0 (camera_lidar_semantic).
```

Performing retention analytics on IBM Storage Protect archive data

Use IBM Data Cataloging to perform retention analytics on archive data managed by IBM Storage Protect.

Before you begin

You must have an admin role to do this task.

About this task

IBM Data Cataloging feature is used to perform search, analytics, and reporting on the retention period for archive data managed by the IBM Storage Protect.

The existing IBM Storage Protect data connection and scan function in the IBM Data Cataloging is enhanced to obtain the additional retention information from the IBM Storage Protect servers for the archive data.

Grouping expiration values to tag values
: You can group the expiration values and run SQL queries to track the retention time period for the archive data managed by IBM Storage Protect. The custom tags are applied by `AUTOTAG` policies that look for all the data that is about to expire by a certain time period. The tagged policies can be scheduled to run periodically. After you apply `AUTOTAG` policies, the IBM Data Cataloging GUI and REST APIs can be used to search, discover, and perform analytics on the data that is going to expire during the various time periods.

Perform the following steps to apply retention analytics on the IBM Storage Protect archive data.

Procedure

1. Enable retention analytics feature. For more information, see, Enabling the retention analytics feature.
2. Add IBM Storage Protect data source and initiating a scan. For more information, see Adding IBM Storage Protect data source and initiating scan.
3. Search the expiry time. For more information, see Searching expiry time.

- **Enabling the retention analytics feature**
  Enable the retention analytics feature in IBM Data Cataloging.
- **Adding IBM Storage Protect data source and initiating scan**
  Add IBM Storage Protect data source before you initiate a scan on the archive data.
- **Searching expiry time**
  Search the expiration time for the IBM Storage Protect archive data that is going to expire in certain time.

Enabling the retention analytics feature

Enable the retention analytics feature in IBM Data Cataloging.

About this task

To enable the retention analytics for IBM Storage Protect archive data feature in IBM Data Cataloging, you must first set an environment variable in IBM Data Cataloging.

Following are the list of environment variable that you can set in the IBM Data Cataloging. The expiration data is not scanned if the expiration variables are not set.

PROTECT_ARCHIVE_WITH_EXPIRE_ONLY
: When the variable is set to 'True', the IBM Storage Protect scan feature scans the archive data only and includes the expiration data that is needed for the expiration date analysis. When the variable is set to 'False' or not set at all, the IBM Storage Protect performs the default action of scanning all the backup and archive records without including the expiration data that is needed for the expiration date analysis.
Important: When the variable value is set, the IBM Data Cataloging server performs IBM Data Cataloging scans on the archive data only that includes the expiration data. This action takes precedence over any other environment variables that are defined for the retention analytics feature.

PROTECT_ARCHIVE_EXPIRATION
: When the variable is set to 'True', IBM Storage Protect scans the archive data and checks that expiration data is also scanned. When the variable is set to 'False' or not set at all, the expiration data is not included, unless the value of *PROTECT_ARCHIVE_WITH_EXPIRE_ONLY* is set to true.

PROTECT_ARCHIVE_DISABLED
: When the variable value is set to 'True', IBM Storage Protect scan feature does not scan the archive data unless the value of *PROTECT_ARCHIVE_WITH_EXPIRE_ONLY* is set to true. When set to 'False' or not set at all, IBM Storage Protect scan feature scans the archive data.

PROTECT_BACKUP_DISABLED
: When the variable is set to 'True', IBM Storage Protect scan feature does not scan the backup data. When set to 'False' or not set at all, IBM Storage Protect scan feature scans the backup data unless the value of *PROTECT_ARCHIVE_WITH_EXPIRE_ONLY* is set to true.

Important: The *PROTECT_ARCHIVE_EXPIRATION* value must be set to "True", if a complete scan of the IBM Storage Protect connection with the archive retention feature is to be completed.
Perform the following steps for enabling retention analytics feature in the IBM Data Cataloging environment:

Procedure

1. Issue the following command to log in to the IBM Data Cataloging by using ssh:

   ```
   ssh <adminuser>@<data_cataloging_host>
   ```

   Note: The `<adminuser>` is a IBM Data Cataloging server user who is assigned with an administrator role and the `<data_cataloging_host>` is either the FDQN or IP of the IBM Data Cataloging server.
2. Issue the following command to edit the configmap:

   ```
   oc edit configmap connmgr
   ```

   After you run the preceding command, the output looks similar to the following as shown here. You can modify the following file and add or update the variables and its value in the `data` section.

   ```
   apiVersion: v1
   data:
     CONNMGR_ENDPOINT: /connmgr/v1/
   ```

```
  CONNMGR_PROTOCOL: http
  PROTECT_ARCHIVE_WITH_EXPIRE_ONLY: "True"
kind: ConfigMap
metadata:
  creationTimestamp: "2022-08-26T19:55:39Z"
  name: connmgr
  namespace: spectrum-discover
  ownerReferences:
  - apiVersion: spectrum-discover.ibm.com/v1alpha1
    kind: SpectrumDiscover
    name: spectrumdiscover
    uid: 001fb331-c2ef-4b28-8827-5c7d6a702904
  resourceVersion: "13761646"
  uid: bf621847-4c00-44c9-a7f2-4d3430ede67b
```

3. Save the preceding updated file and issue the following command to verify that the data is updated.

   ```
   oc get configmap connmgr -o yaml
   ```

4. Issue the following commands to verify that the update is made in the connection manager pod:

   ```
   oc -n ibm-data-cataloging delete sts/isd-connmgr
   ```

Adding IBM Storage Protect data source and initiating scan

Add IBM Storage Protect data source before you initiate a scan on the archive data.

About this task

After the retention analytics feature is enabled in the IBM Data Cataloging, you can add one, or more IBM Storage Protect data sources and initiate a scan. For more information, see Creating an IBM Storage Protect data source connection.

Important: After the scan is complete, the database needs to be refreshed before you add the tags and the policies.
Do the following to tag expiry time:

Procedure

1. Log in to the IBM Data Cataloging GUI.
2. Click  and go to Metadata > Tag Management
3. On the Tags tab, click the Add Tag.
4. In the New organizational tags window, enter the tag name in the Name field.
   For example, enter `ExpirationDate`, `ExpirationDays`, or `Expiration` as a tag name.
5. Select `Restricted` from the tag Type list.

   Restricted
   : A Restricted tag can be anything that describes groups of records, but is restricted to a set of pre-defined values, such as data classification or billing department number.

6. Enter the expiration values into the Values field and press Enter key to save each expiration value.
   For example, to group all the data that is going to expire by the end of a year, create a tag that is called 'Expiration Year'. Similarly, to group all the data that is going to expire after few days from now, create a tag that is called 'ExpirationDays'.
   Each saved expiration values (date or number of days) are grouped and displayed underneath the Values field. These expiration values are grouped and associated with a custom tag that is created. For example, tag values can be 01-January-2021 for a date-specific expiration or, 1 day, 1 month, and 1 year for days-specific expiration.
7. Click Submit.
   The tag name, types, and values are displayed on the Tags tab after you submit the record.
   Important: You must add all the expected values before you submit the record.
8. Go to the Policies tab and click Add policy.
9. On the Define page, enter the policy name in the Name field.
10. Select `AUTOTAG` from Policy Type list.
11. On the Policy page, enter the values in Filter field.

    Sample filter input for 30-days from today as expiration period
    ```
    mtime > NOW() AND mtime < (NOW() + 30 DAYS) AND platform like 'Spectrum Protect' and state like 'Archive'
    ```
    Sample filter input for all data that is going to expire by the end of year 2021
    ```
    State in 'Archive' AND mtime between '2021-01-01' and '2021-12-31'
    ```
    Important: The date in the query must be in the format of 'YYYY-MM-DD'.

12. Choose the custom tag created to track the expiration time period from Tag list.
    The tag list contains all the custom tags created with their saved tag name, type, and value details created on Tags tab. You can add more than one tag here to tag a policy.
13. Choose the expiration value (date or number of day's) from the Values list.
    The expiration values that you choose from the Values list are tagged with custom tag that you choose from the Tag list.
    For example, if you choose `ExpirationDays` as custom tag, you can choose the expiration value `30DayExpiration` as the number of days for data to expire to tag with the custom tag. Alternatively, if you want to group records by all the data that is going to expire by the end of a year, select the 'ExpirationYear' value.

14. Schedule the policy to run periodically by using the available schedule options.

    Schedule options
    : Now
      Daily

Weekly
Monthly

15. Verify the policy details on Review page before you submit the save the policy.

    Sample policy detail on review page as shown

```
Name:
30DayExpirationPolicy
Action id:
AUTOTAG
Pol filter:
mtime > NOW() AND mtime < (NOW() + 30 DAYS) AND platform like '%Spectrum Protect%' and state like '%Archive%'
Pol state:
active
Collections:
```

Searching expiry time

Search the expiration time for the IBM Storage Protect archive data that is going to expire in certain time.

About this task

After the IBM Storage Protect archive record expiration dates are tagged, do the following to search and apply further analytics on the records.

Important: Each policy that is defined must be executed before you perform search.

Procedure

1. Log in to the IBM Data Cataloging GUI.
2. Click  and go to Search > Query builder > SQL query page.
3. Choose groupings based on the tag that is created from Group by list.
   For example, ExpirationDate.
4. Enter the SQL query in the query box.
   For example, `State in 'Archive' AND mtime between '2021-01-01' and '2021-12-31'`
   Important: The mtime date in the query must be in the format of 'YYYY-MM-DD'.
5. Set the limit for the number of records returned from the Limit results list.
6. Click View results.
   The Query Results page displays the grouped results with expiration time, total files filtered, and the total size.
   Note: Where, Total size represents the original size of the files that are stored in the backend. Total size that is returned in the report does not account for space savings from compression and data deduplication. Therefore, can over-state the amount of storage that is going to expire on the IBM Storage Protect server. Currently, the data expiration support within the IBM Data Cataloging does not consider the amount of space that is saved in the size of the file for container pools that supports compression or data deduplication.
7. To generate a report, select the checkbox and click Generate report. On the Generate report dialog, Enter the report name in the Name field, and click Submit.
8. To view the report, go to Search > Reports.
   After you submit the report, the new report gets generated and listed on Reports page in a table with following metadata details:
   - Report name
   - Last run
   - Duration
   - Status
   - Output size
9. Select a report from the table lists and click Report summary.
   A View Data Report window is displayed with all the details of a report.

   Sample report details as shown

```
View Data Report

Report
SProtectDemoReport
Last Run
Wed Jun 30 2021 19:09:52 GMT+0530 (India Standard Time)
Duration
0 seconds
Status
Complete
Size
247 Bytes
Name
"SProtectDemoReport"
Group By
["TAG7","datasource","filespace","nodename"]
Filters
[]
Sort By
""
Query
"(daystoexpire = '180' AND datasource = 'WIN-RUT6M1TE74U' AND filespace = '\\\\win-rut6m1te74u\\c$' AND nodename =
'WIN-RUT6M1TE74U') OR (daystoexpire = '720' AND datasource = 'WIN-RUT6M1TE74U' AND filespace = '\\\\win-
```

```
rut6m1te74u\\c$' AND nodename = 'WIN-RUT6M1TE74U')"
See on table
```

Managing tags

A tag is a custom metadata field used to enhance system metadata with organization-specific information. For instance, an organization may categorize its storage by project, cost center, or region but the attributes not provided by the storage system itself. Tags enable data classification, search, and reporting based on these organizational dimensions. Use tags to store this additional information and to manage or locate data using metadata that reflects your organizational context.

Permissions

Security Administrators
: Cannot create, update, delete or list any type of tag.

Data Administrators
: Can create, modify, delete, and list both `Label` and `Tag` tags.

Data Users
: Can list all tags.
Cannot create, modify or delete `Tag` tags.
Can create and modify `Label` tags.

Types of tags

Label
: Lightweight identifiers used for simple tagging. Useful for short descriptors like `project`, `region`, or `team`. Value size can be up to `256`, `1024`, or `4096` characters.

Tag
: Structured tags with a predefined list of accepted values. Ideal for controlled vocabularies such as `status`, `compliance level`, or `department`. Value size is always limited to `4096` characters.

- **Creating tags**
  Use the following information when you create tags.
- **Viewing and searching tags**
  View and search tags from the Tags tab on the Metadata page to locate specific tags by name, type, or value, and sort or filter results as needed.
- **Editing tags**
  Edit tag values from the Tags tab on the Metadata page to update the allowed values for existing organizational tags in IBM Spectrum Discover.
- **Deleting tags**
  Delete tags from the Tags tab on the Metadata page to remove metadata classifications that are no longer needed from the IBM Spectrum Discover catalog.

Creating tags

Use the following information when you create tags.

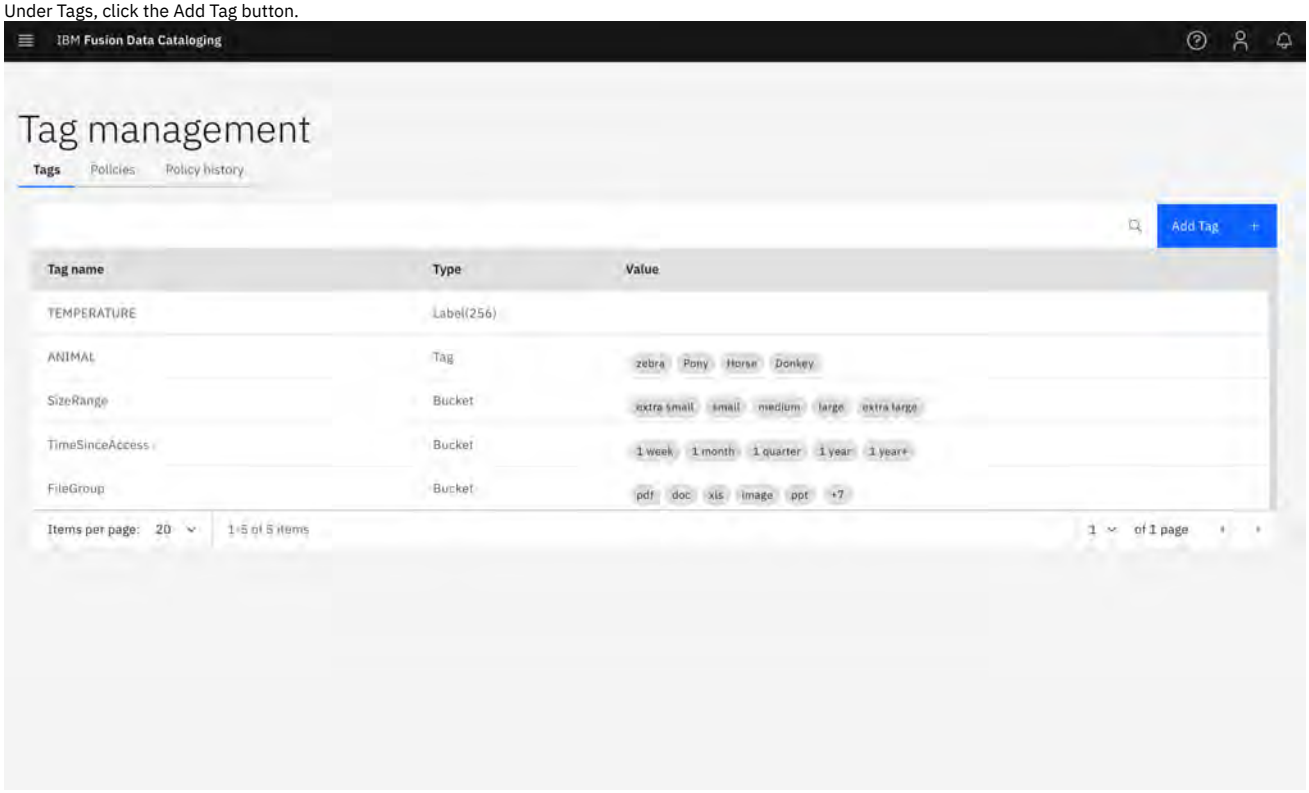
About this task

Use the Tags tab to create new tags. The table lists the tag name in the Field Name column, the Type, and the tag values in the Tags column. Use the icons to Edit or Delete a tag.

Procedure

1. Go to Metadata > Tag Management

2. Under Tags, click the Add Tag button.

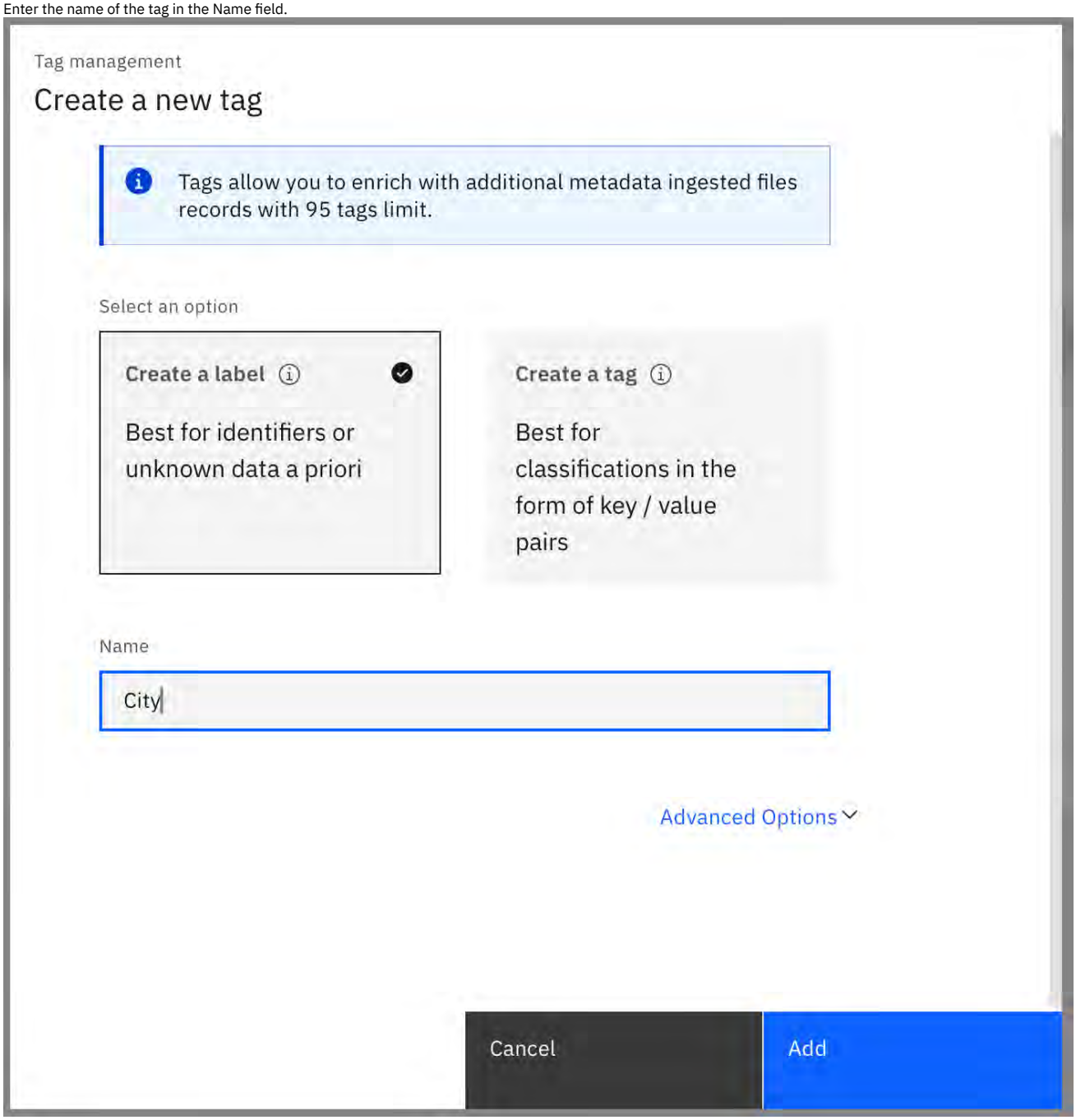


3. In the Create a new tag page, select the type of tag you want to create.

Label
: A lightweight tag used to describe records with simple identifiers such as project name, department, or device ID. Labels are not restricted to predefined values unless explicitly configured.

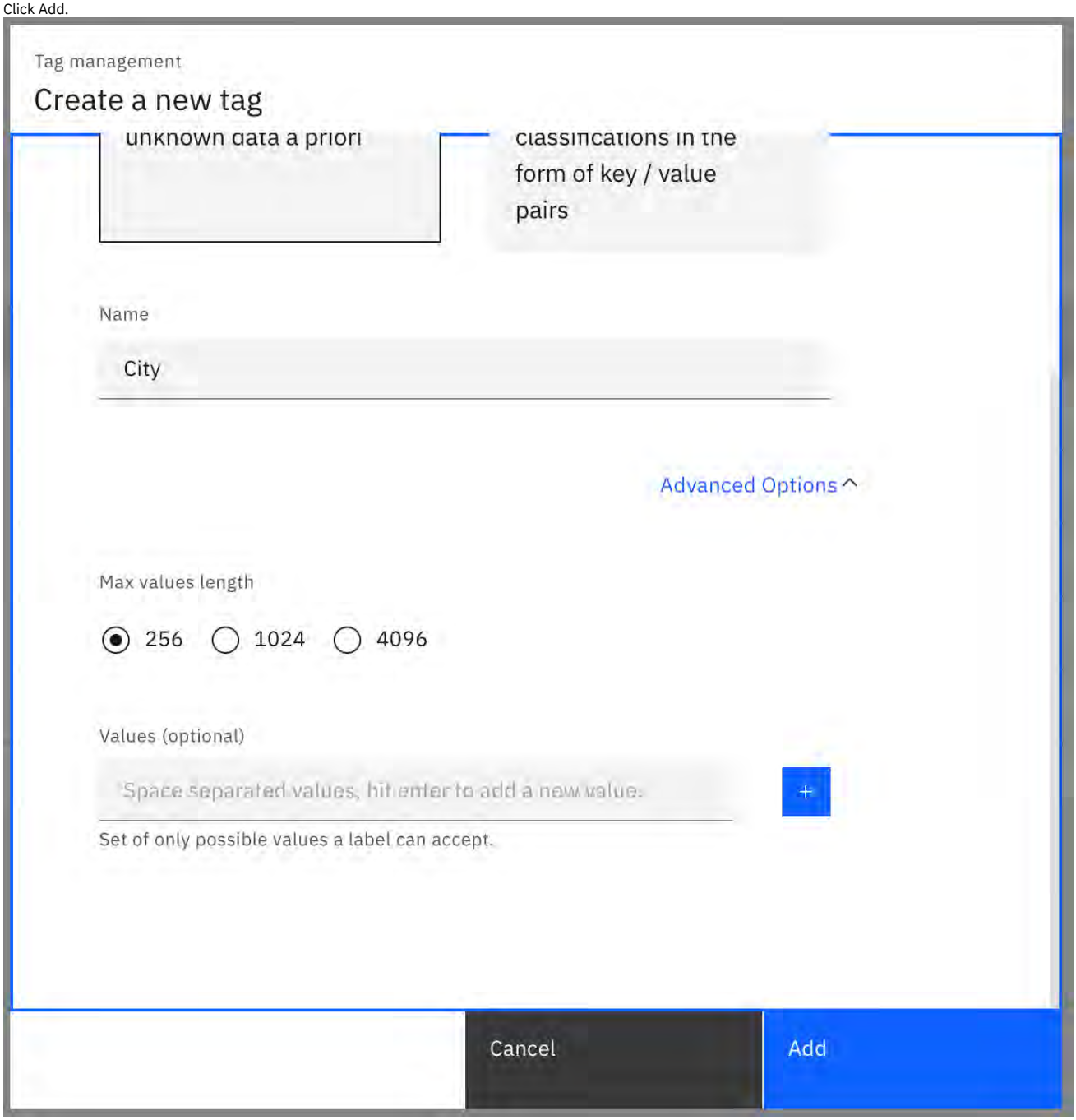
Tag
: A structured tag that enforces a predefined set of accepted values. Useful for controlled vocabularies like classification levels or billing codes.

4. Enter the name of the tag in the Name field.



5. Optional: If the tag type is Label, then click Advanced Options you can select a maximum value length of `256`, `1024`, or `4096` characters.
6. Define the following values for the tag or label:
   - These values are only accepted inputs when the tag is applied to metadata.
   - Enter one or more values into the **Values** box.
   - Press the Enter key after each value. Saved values appears in the input box.
     Note: The total combined length of all values must not exceed `1024` characters.

7. Click Add.



The tags, their types, and values are displayed in the table on the Tags tab.

Viewing and searching tags

View and search tags from the Tags tab on the Metadata page to locate specific tags by name, type, or value, and sort or filter results as needed.

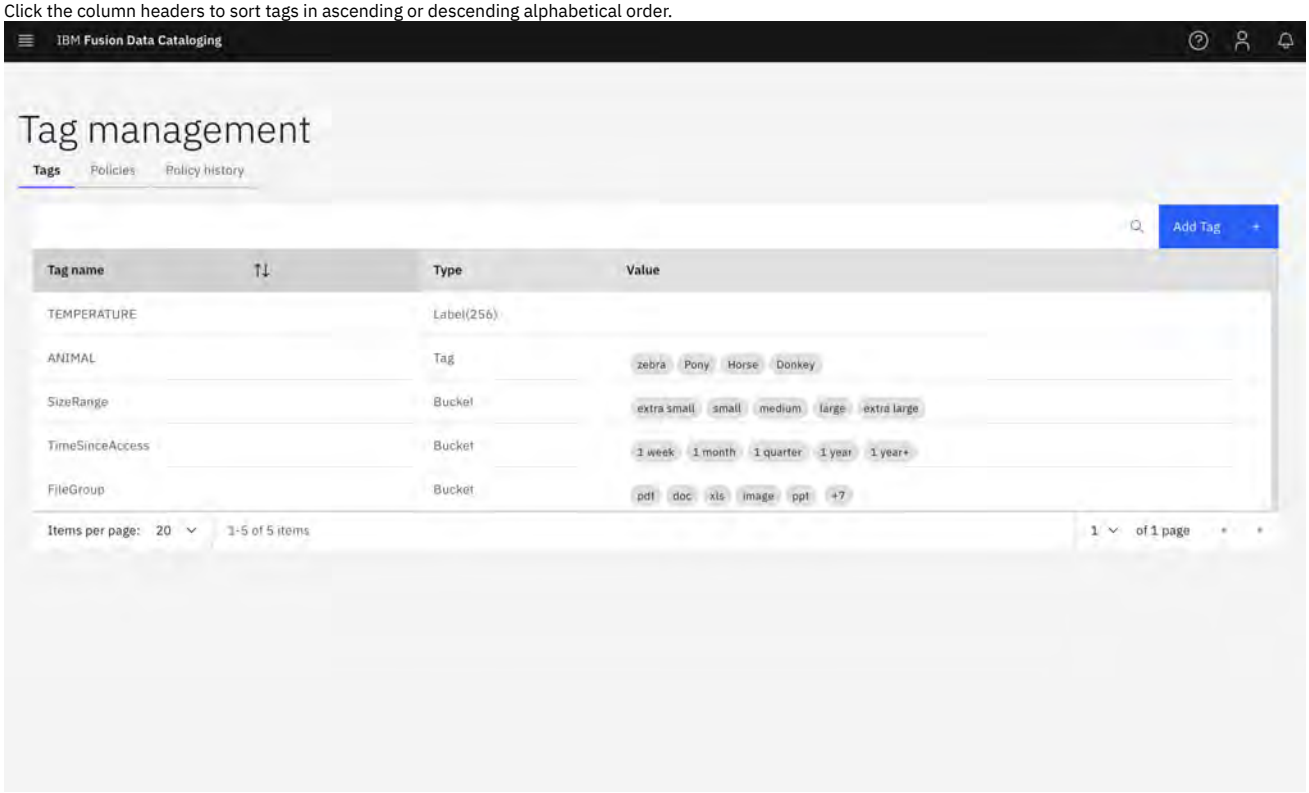
About this task

You can see a list of all tags or search for a specific ones from the Tags tab on the Metadata page.

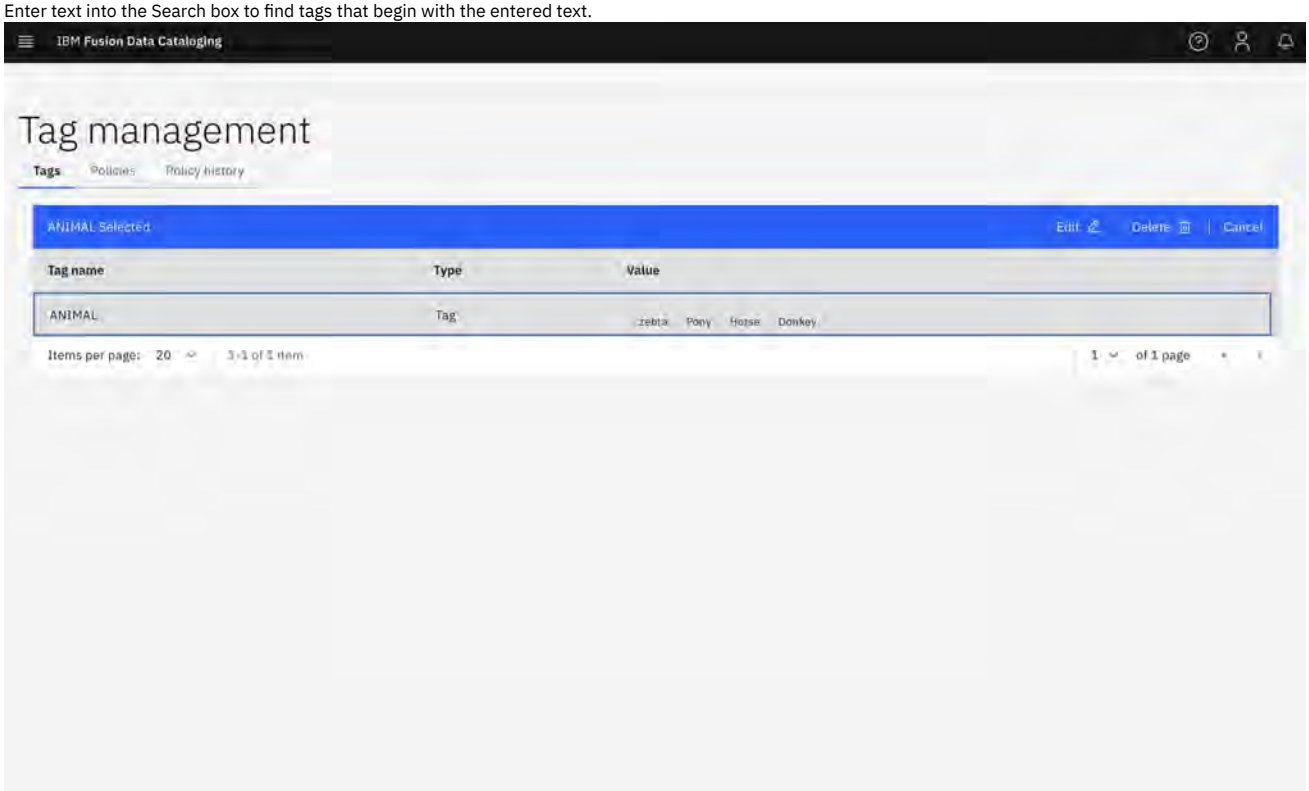
Procedure

1. Go to Metadata > Tag Management.
2. Under Tags tab, a list of tag **Names**, **Types**, and **Values** are displayed.

3. Click the column headers to sort tags in ascending or descending alphabetical order.



4. Enter text into the Search box to find tags that begin with the entered text.



As you type, a filtered list of tags containing the text string is automatically displayed.

5. Select a tag and click Edit or Delete from the header row to perform the desired action.

Editing tags

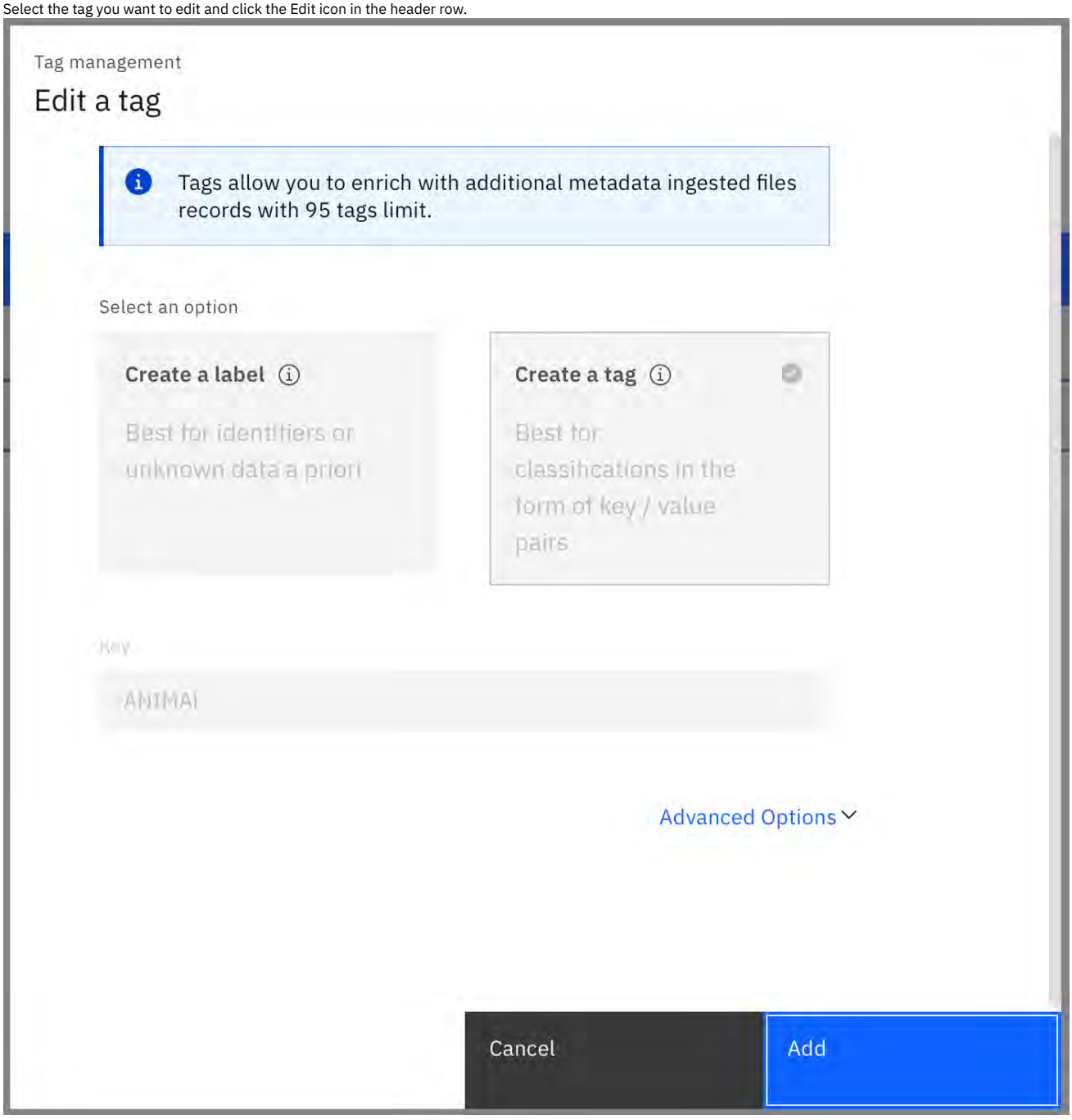
Edit tag values from the Tags tab on the Metadata page to update the allowed values for existing organizational tags in IBM Spectrum Discover.

About this task

You can edit tag values from the **Tags** tab on the **Metadata** page.

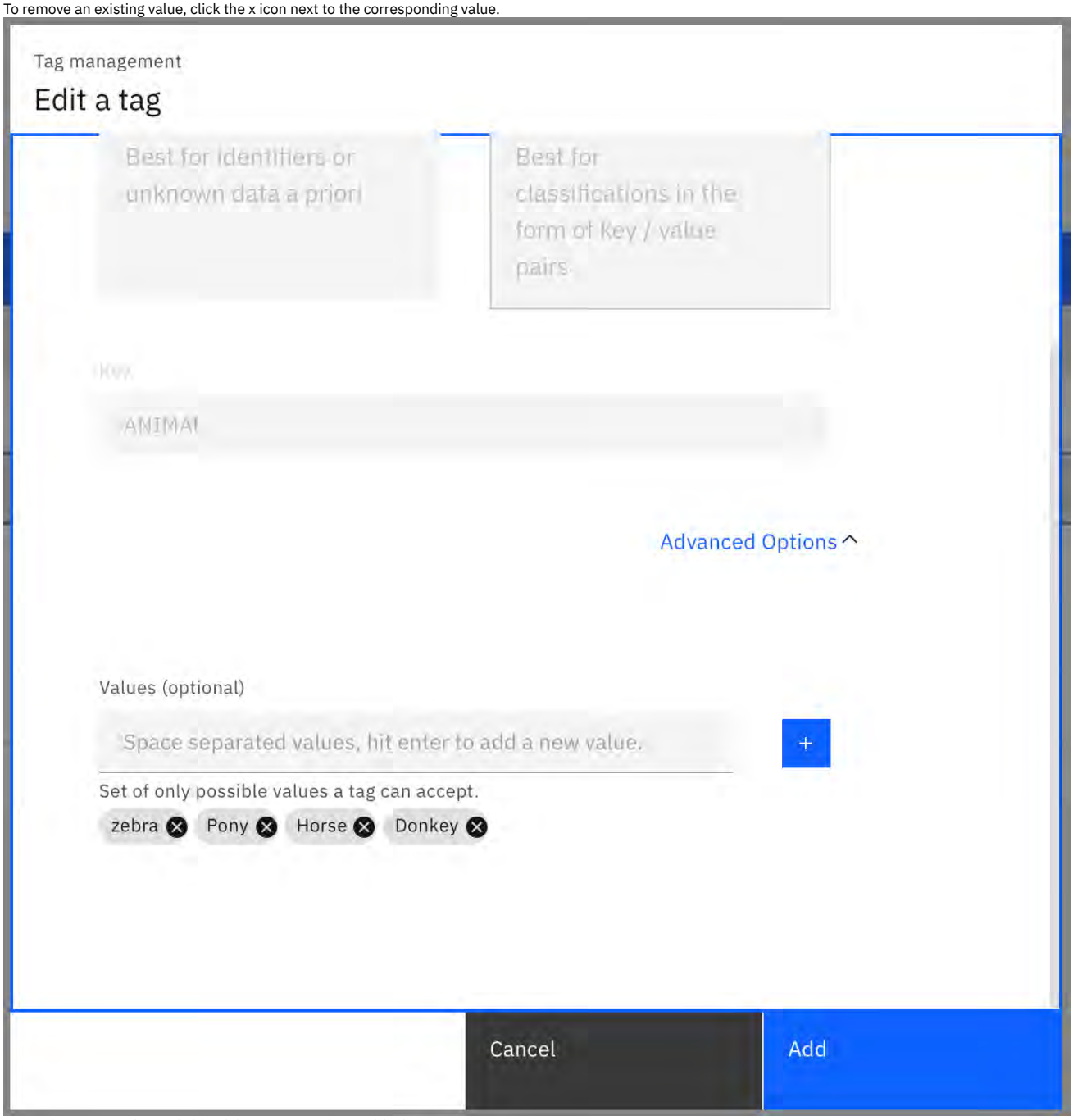
Procedure

1. Go to Metadata > Tag Management.
2. Select the tag you want to edit and click the Edit icon in the header row.



3. In the Edit a tag window that appears, enter one or more new values in the Values field and click Enter.
   Saved values are displayed in the input box.

4. To remove an existing value, click the x icon next to the corresponding value.



Note: The Name and Type fields of the tag cannot be modified.

5. Click Submit.
   The modified values are listed in the table under the Tags tab.

Deleting tags

Delete tags from the Tags tab on the Metadata page to remove metadata classifications that are no longer needed from the IBM Spectrum Discover catalog.

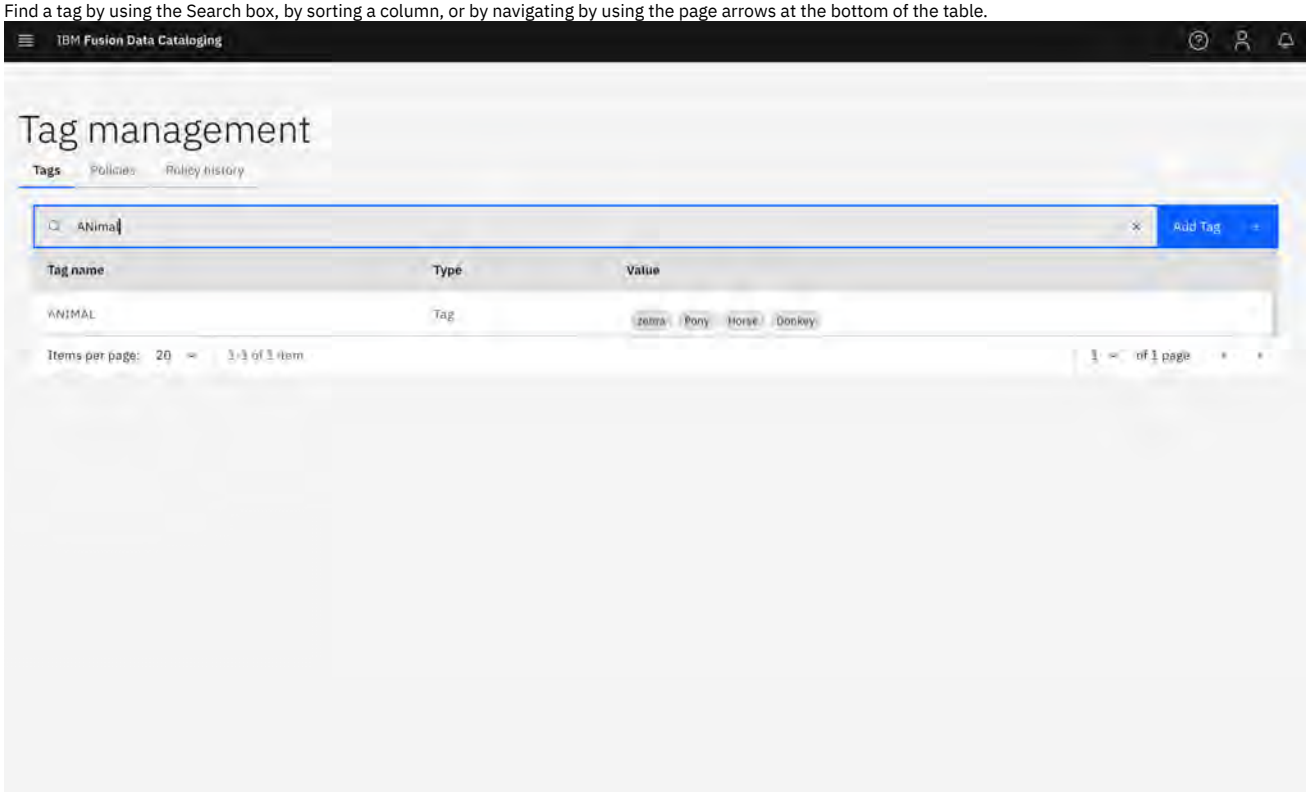
About this task

You can delete tags from the Tags tab on the Metadata page.

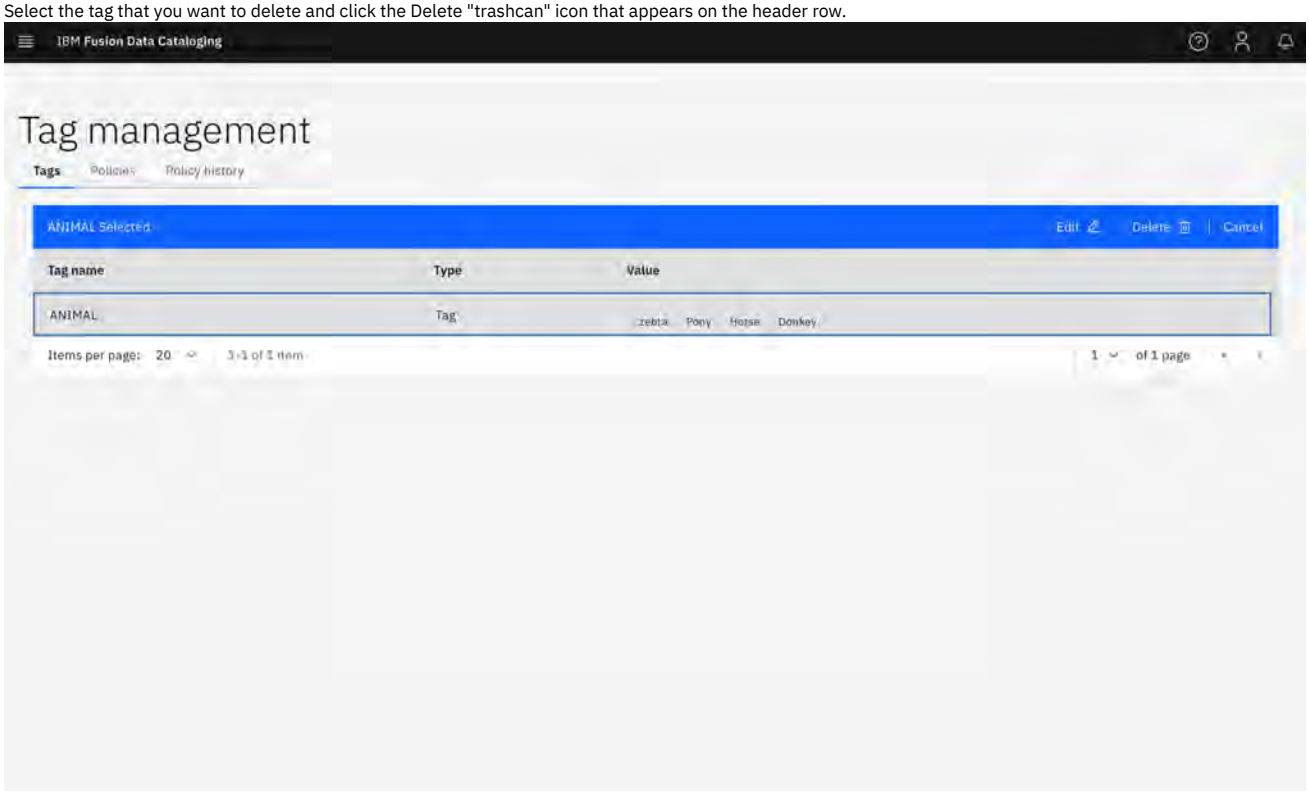
Procedure

1. Go to Metadata > Tag Management.

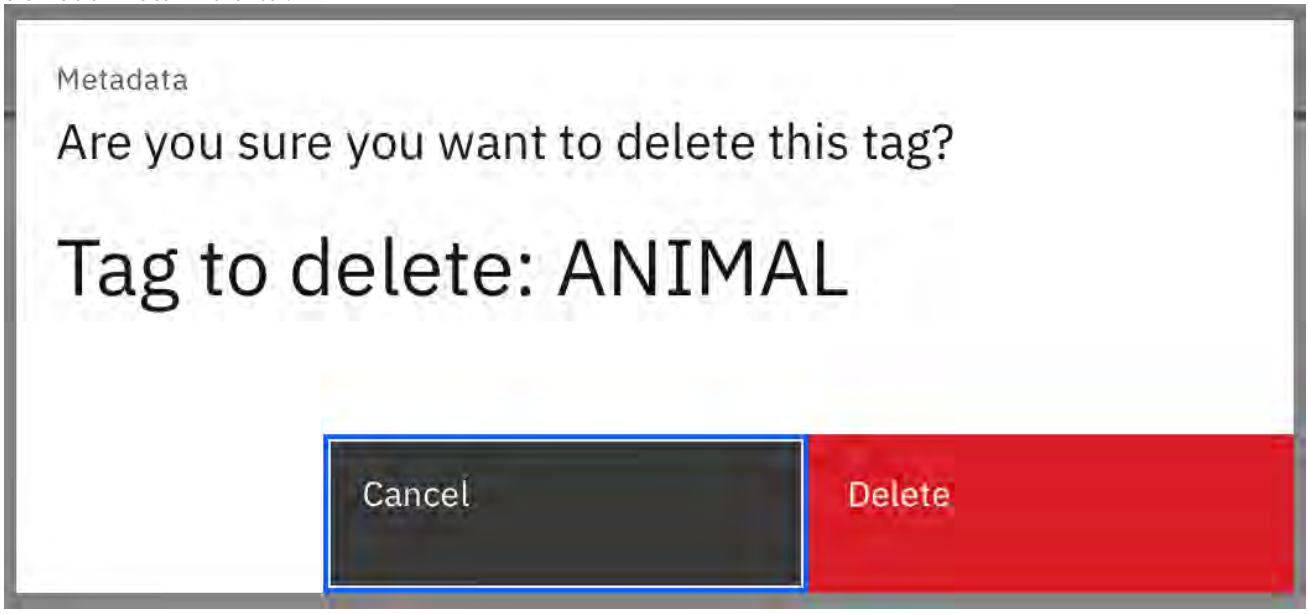
2. Find a tag by using the Search box, by sorting a column, or by navigating by using the page arrows at the bottom of the table.



3. Select the tag that you want to delete and click the Delete "trashcan" icon that appears on the header row.



4. Click Delete in the confirmation box.



The tag is removed from the table in the Tags tab.

Discover data

By discovering your data, you can apply policies that assign tags to your data. You can apply tags to the results of a single search, or you can use policies to automatically apply tags on a periodic basis.

There are three ways to discover data:

- Content-based keyword and tagging. The search is based on regular expression patterns that are defined within IBM Data Cataloging. For more information, see Creating a content search policy.
- Create a policy by using tags with known values. A policy is automatically run against all data that meets criteria that are specified in a filter. For more information about creating and using policies, see Managing metadata policies.
- Search your data by using a query in standard SQL grammar or do a visual exploration of tags by point-and-click. For more information, see Searching system and custom metadata fields.

- **Searching**
  Build visual or SQL queries in the Query Builder to search for data in IBM Spectrum Discover, then filter, sort, and group the results to drill down to the relevant records.
- **Grouping data by file type**
  Create policies to group data by file type.
- **Searching system and custom metadata fields**
  Search system and custom metadata fields using SQL queries in the Query Builder to filter, sort, and retrieve file records from the IBM Spectrum Discover data catalog.

Searching

Build visual or SQL queries in the Query Builder to search for data in IBM Spectrum Discover, then filter, sort, and group the results to drill down to the relevant records.

About this task

You can build a visual query or a custom SQL query to begin the search. In a visual query, you can add tags from the search page and view the values in the form of charts, tables, or directories for a more enhanced experience. The following procedure helps you build queries to search for relevant results.

Procedure

1. Click  on the upper left of the IBM Data Cataloging interface and then click Search.
   The Query Builder page appears.
2. To build a visual query, click Add corresponding to the Tag that you want to include in your search.
3. On the Select Values page, select the values in the table that is displayed beneath the chart to build your query.
4. Click Update Query to preview your query on the Query Builder page.
   Note:
   - The Potential Record Count under Query Summary reflects the expected count of records in the search result that is displayed for the selected tags.
   - Click Reset Query to delete your current selection and build the query again.
   - Click  on the Query Preview box to copy the icon for future use.
5. Click View Results to display the search results.

6. To build a custom query, click the SQL Query tab on the Query Builder page.
7. Type your query in the text area provided.
   Note:
   - Click the + icon on the right to select from the query field suggestions.
   - Click  to view all recent search queries that were previously used.
8. Click View Results to display the search results.
   On the Query Result page, the search results are displayed under the Individual tab.
9. Select the tags that are provided under the Grouped By list on the left to drill down the results further.
   The Query results are now displayed under the Condensed tab.
   Note:
   - Click  to edit the displayed columns from the list that appears.
   - Select a search result row and then click Generate Report to generate a report for the set of records.
   - Select a search result row and then click View Individual Records to view the records with no grouping selected.
   - Click Back to query builder to return to the query builder screen from any search query pages.

Grouping data by file type

Create policies to group data by file type.

About this task

You can create AUTOTAG policies and run them on a periodic basis to group data by specific file types.

Procedure

1. Create an Open tag to group data. For example, FileTypeGroup.
   a. Go to Metadata > Tag Management
   b. Under Tags click Add Tags.
   c. Select `Open` in the Type menu.
   d. Click Submit.
2. Create an AUTOTAG policy with a relevant filter criteria. For example, `Filetype = 'pdf'`.
   a. Go to Metadata > Tag Management.
   b. Under Policies click Add Policy.
   c. Enter a name for the policy in the Name box. For example, `FileTypeGroupTagging`.
   d. Select `Autotag` from the Policy Type menu.
   e. Click Next step.
   f. Type `filetype = 'pdf'` as the filter criteria.
      Note: You can modify the filter criteria based on the file type you want to group. For example, if you want to group all JPG images, type `filetype = 'jpg'`.
   g. Click Add Tag.
   h. Select the file type group tag that you previously defined. In this example, the tag name is `FileTypeGroup`.
   i. Match the tag value with the filetype in the filter criteria. For example, if the filter criteria is `filetype = 'pdf'` use `pdf` as the tag value and then click Next step.
   j. Click the slider control to set the policy status to `active`.
   k. Under Schedule select the frequency for running the policy and then click Next step. For example, Now, Daily, Weekly, Monthly
   l. Review the policy and click Save to run the AUTOTAG policy.

   Note: You can manually refresh the IBM Data Cataloging database to confirm whether the policy is run properly. Refreshing the database is optional and not an essential prerequisite for the process. To manually refresh the database, follow the procedure shown:
   a. Go to Data Connections > Discover Databases.
   b. Click Run table refresh under the Metadata Summarization database
3. Create a policy to group data with a different file type tag value or modify the existing policy.
4. Define a schedule to rerun policies on a periodic basis.

Searching system and custom metadata fields

Search system and custom metadata fields using SQL queries in the Query Builder to filter, sort, and retrieve file records from the IBM Spectrum Discover data catalog.

1. On the Query Builder page, click SQL query.
   a. Enter your query directly and click View results. Use the standard grammar that is used in an SQL query.
   b. Click + icon to select a suggested field from the list that appears, complete your query, and click View results.
   c. Click  and select from the search queries that were previously used.
   d. Modify the query if necessary, and click View results.

   The query language is SQL. The underlying code takes care of certain semantics, for example,
   - Keyword
   - Columns to select
   - Name of the databases
   - Where clause
   - Limits
   - Offsets
   - Order by clauses

   The search clause that is input by the user is only the body of the query that would appear after the where clause and before the limit/offset/order qualifiers.

- **System metadata fields to search on**
  The list in this section provides definitions of items on which you can search system metadata fields.
- **Access control list metadata to search on**
  Access Control List (ACL) metadata is collected from SMB/CIFS data source search results.
- **Search on custom metadata fields**
  You can do a search on custom metadata fields.
- **Examples of search filters**
  This section provides a list of examples for search filters.
- **Search results table**
  The search results table displays information about the records that met the search criteria.
- **Refine search results**
  After a set of search results is returned, you can further refine the data by using the following options:
- **Sort search results**
  You can sort search results by column.
- **Tag search results manually**
  After a filtered set of records is displayed on the search pane, you can select all or some of those documents and apply organizational tags to them.

System metadata fields to search on

The list in this section provides definitions of items on which you can search system metadata fields.

The following list shows search filters that you can use for a search.

Datasource
: The name of the datasource where the record originated. The datasource refers to the label of the source storage system that was defined in the IBM Data Cataloging connection management panel.

Platform
: The type of storage system from which this record originated.

Site
: The physical site for the data as input by the user at scan time.

Cluster
: The name of the IBM Storage Scale cluster to which the record belongs. This term applies only to IBM Storage Scale.

NodeName
: For IBM Storage Protect, this indicates the node or client system to which the backup or archive record belongs.

Fileset
: The file set to which the record belongs for IBM Storage Scale. This term applies only to IBM Storage Scale.

MgmtClass
: For IBM Storage Protect, this indicates the management class to which the backup or archive record belongs.

Owner
: The system metadata owner of the record (file only).

Group
: The system metadata group owner of the record (file only).

UID
: The numeric ID of file owner (file only).

GID
: The numeric ID of file group (file only).

Path
: The file path or object storage bucket of the file that is represented by this record.

Filename
: The name of the file or object represented by the record.

Filespace
: For IBM Storage Protect, this indicates the file space to which the backup or archive record belongs.

Filetype
: The type of the file or `object.MtimeLast` modified time for the file (file only).

State
: The state of the file or object. Possible values for IBM Storage Scale are:

  premig
  : Premigrated

  migrtd
  : Migrated

  resdnt
  : Resident

  Possible values for IBM Storage Protect are:

  ACTIVE
  : Active backup copy

  INACTIVE
  : Inactive backup copy

  ARCHIVE
  : Archive copy

Mtime
: Last modified time for the file (file only).

Atime
: Last accessed time for the file (file only).

Time

Creation time of the file (file only).

Size
: Size of the file or object.

Inode
: The inode of the file (file only).

Permissions
: The permissions of the file (file only).

sizeConsumed
: The size of the consumed capacity (file only).

recordversion
: The version enabled for the cloud storage system buckets or vaults. IBM Data Cataloging stores the S3 versioning identifier in the recordversion tag of the corresponding object. Cloud versioning enables you to store multiple copies, or versions, of the same object.

Access control list metadata to search on

Access Control List (ACL) metadata is collected from SMB/CIFS data source search results.

For SMB/CIFS data sources, IBM Data Cataloging collects Access Control List (ACL) metadata in addition to the standard system metadata. This information is stored in two tables, which are the Access Control Owner and Group (ACOG) and the Access Control Entries (ACES).

Important: Since ACL data is not displayed by default and is unavailable using the "visual query", a special SQL query is needed to view ACL data.
Follow the steps to view ACL data for an SQL query:

1. Log in to IBM Data Cataloging GUI.
2. From the main menu, click Search > Query builder
   The Query builder page is displayed.
3. Click SQL query tab.
4. In the Select * from Spectrum Discover where field, use the following query and click View results to view all ACL data.

   ```
   "aces.entrytype like '%ACL'"
   ```

Note:

- To limit the ACL results to NFS or SMB, use the following query:

  ```
  "aces.entrytype = 'NFS4ACL'", "aces.entrytype = 'DACL'", or "aces.entrytype = 'SACL'"
  ```

- For any other specific aces or acog query, use the following query:

  ```
  "aces.accesstype = 'AUDIT'" or "acog.groupname ='UNIX_GROUPS\wheel'"
  ```

For information on how ACL data is used in policies, see How to use ACL data in policies.

Type the criteria in the search bar to search the ACL metadata. Possible fields to search on are:

acog.ownername
: Indicates the owner of the file.

acog.ownerid
: Indicates the security identifier for the owner of the file.

Note: To search for files based on owner name or security identifier, see the following example:

```
acog.ownername='DOMAIN\user'
```

```
acog.ownerid='S-1-1-1-1000'
```

acog.groupname
: Indicates the group of owner of the file.

acog.groupid
: Indicates the security identifier for the group of the owner of the file.

aces.username
: Indicates the user or group name for which this ACE applies.

aces.userid
: Indicates the security identifier for the user or group name for which this ACE applies.

aces.entrytype
: Indicates that the entry type can either be DACL or SACL and NFS4ACL for NFSv4.

aces.accesstype
: Indicates that the access type can be either one of the following options:

  - ALLOWED
  - DENIED
  - AUDIT

Note: To search files with an access control entry that allows everyone to access, see the following example:

```
aces.username='\Everyone' and aces.entrytype='DACL' and aces.access_type='ALLOWED'
```

To search for files that are on a particular data source and display a Deny Access Control Entry, see the following example:

```
datasource in ('smb1') and aces.entrytype='DACL' and aces.accesstype='DENIED'
```

aces.permissions
: Indicates the possible permission levels that include:

  - R - Read
  - W - Write
  - X - Execute
  - D - Delete
  - P - Write access controls
  - O - Owner

  The valid permission combinations are:

  - READ - R + X
  - CHANGE - R + W + X + D
  - FULL - R + W + X+ D + P + O

  For NFSv4 the permissions vary as shown:

  - r - read-data
  - w - write-data
  - a - append-data
  - x - execute
  - d - delete
  - t - read the attributes of the file
  - T - write the attribute of the file
  - n - read the named attributes of the file
  - N - write the named attributes of the file
  - c - read the file ACL
  - C - write the file ACL
  - o - change ownership of the file

  Multiple NFSv4 permissions can be used in combinations like "rxtncy" for every entry.

aces.flags
: Indicates the flags displaying the type of access control entry (ACE).

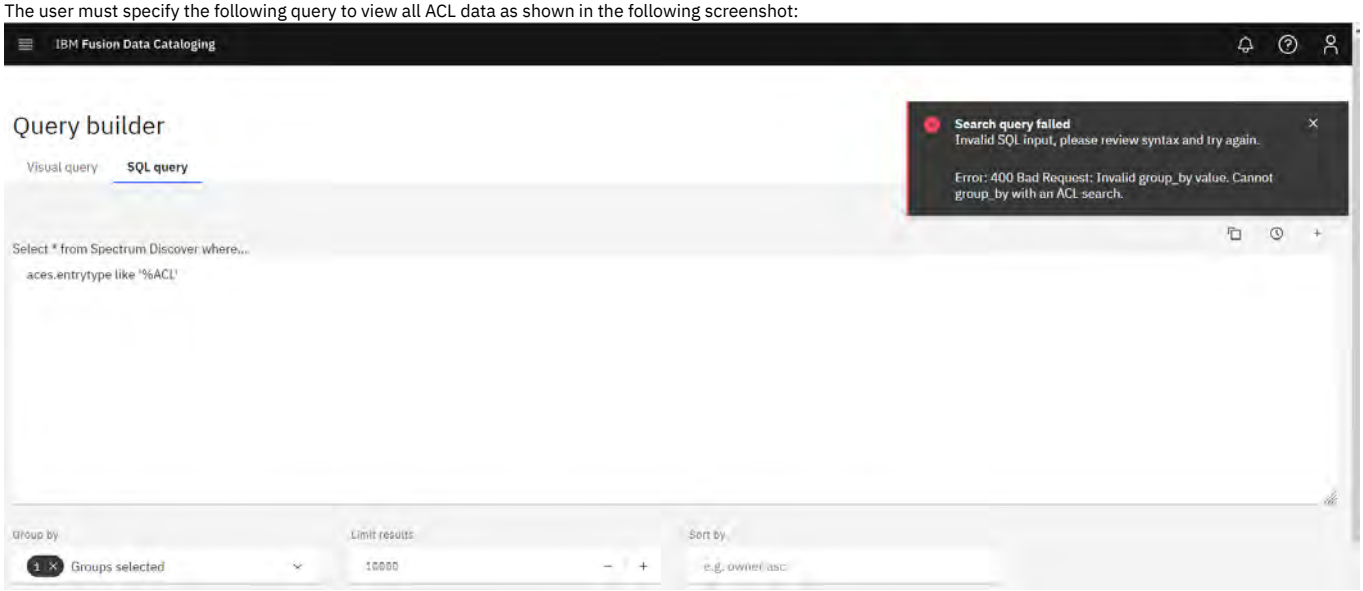
Note: A file can have more than one access control entry (ACE) associated with it. Search results that contain ACL metadata, repeat the file metadata for each ACE. Therefore, reports are the preferred method for using IBM Data Cataloging search results with ACL metadata.

- **How to use ACL data in policies**
  This topic describes how to use ACL data in your policies.

How to use ACL data in policies

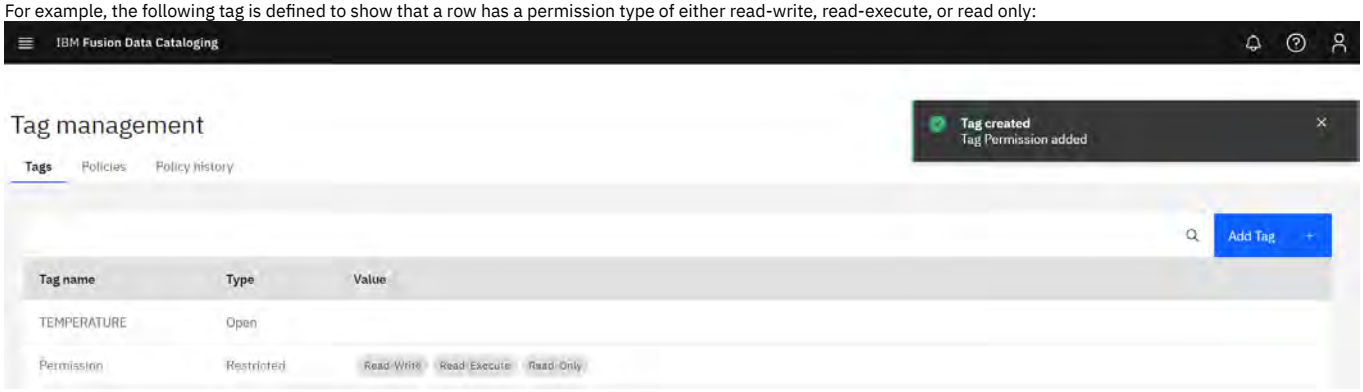
This topic describes how to use ACL data in your policies.

The user must specify the following query to view all ACL data as shown in the following screenshot:

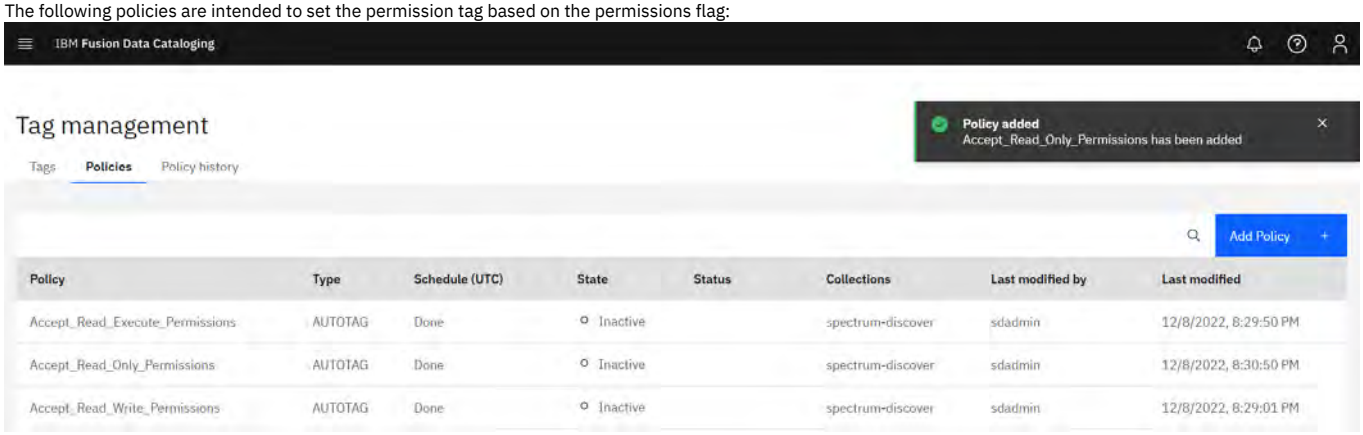


Note: ACL grouping is not supported. To perform such groupings, an external DB client like the DB2 warehouse client should be used against the Spectrum Discover database.
Important: When using acog or aces data to make tagging policies, it is important to understand that a single selection of acog or aces data may result in tags not being meaningful.

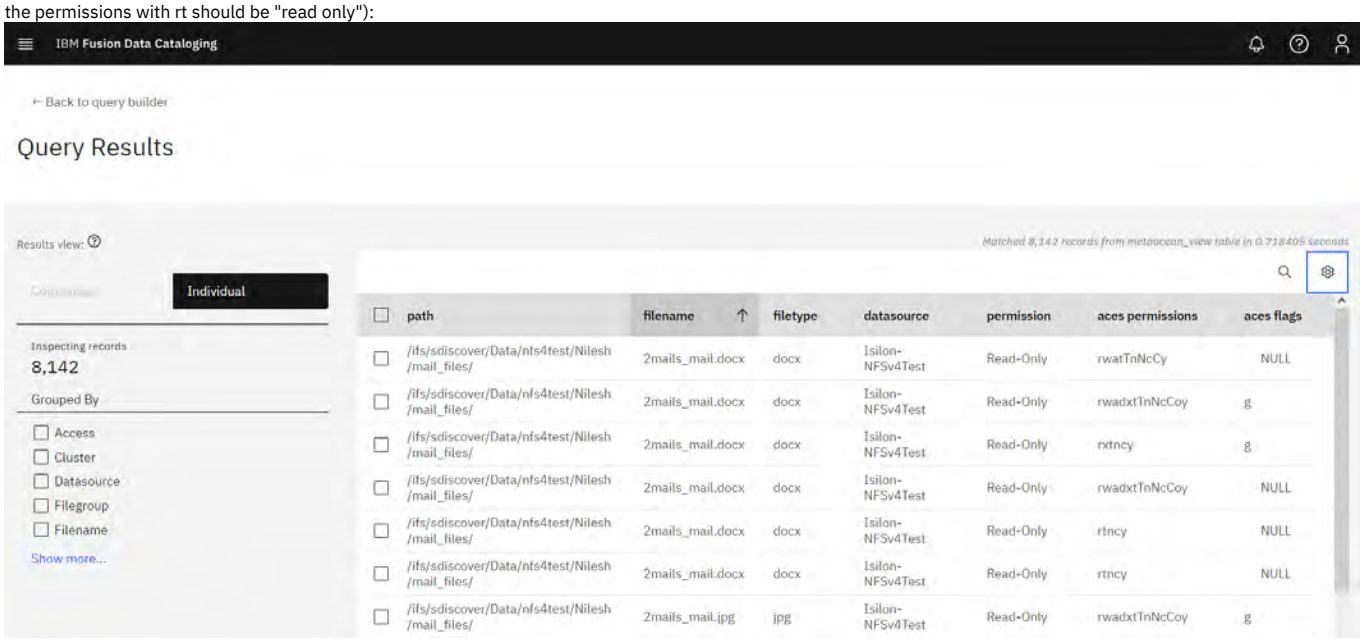
For example, the following tag is defined to show that a row has a permission type of either read-write, read-execute, or read only:



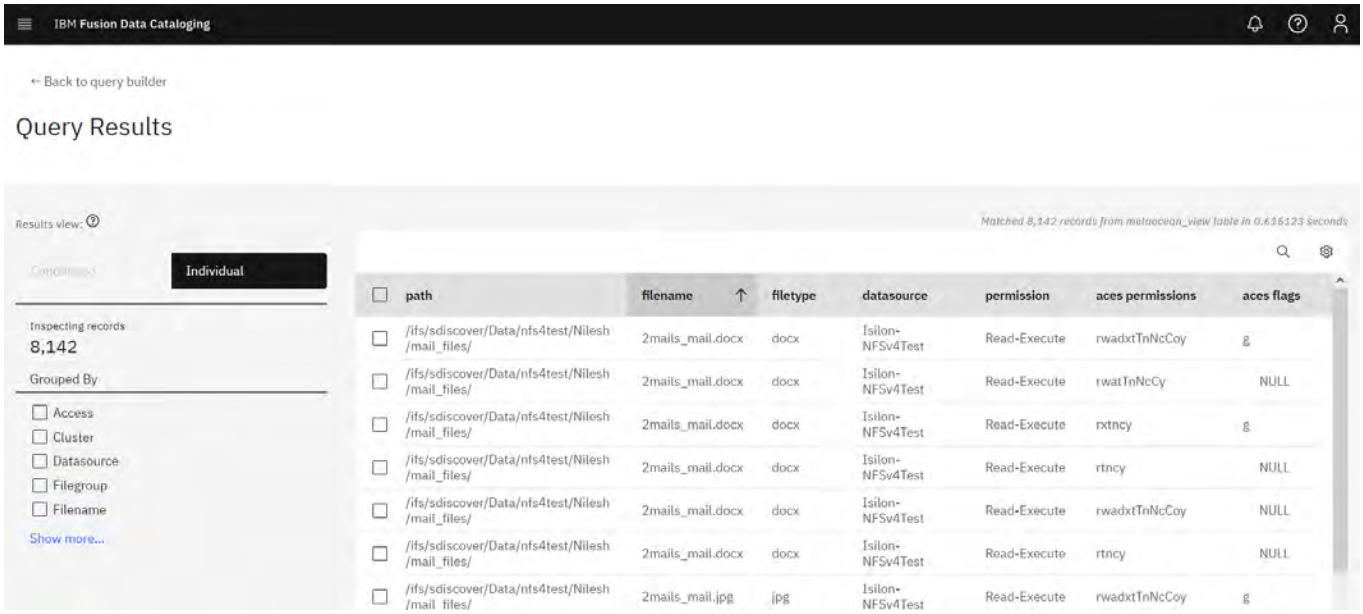
The following policies are intended to set the permission tag based on the permissions flag:



After running the Read Only policy, the Permission tag is populated, but as seen from the following row for the same file, the tag does not match the real permissions (only the permissions with rt should be "read only"):



After running the Read Execute policy, the Permission tag is populated but the permission tag is changed from Read-Only to Read-Execute which makes it meaningless and effectively useless:



To solve this problem, you can create tags and set the value as "True" if the condition exists.
Note: You cannot add a "false" value because if you were to run a policy that found a condition that doesn't exist, you would run into the same meaningless results.
When tags have a "True" value, you can now set the policies to have a "True" value. By doing this, you can determine what permissions are allowed for each file.
Note: You can customize your own tags and use aces/acog information that is meaningful to you.

Search on custom metadata fields

You can do a search on custom metadata fields.

Comparators
To do a search, you can also use the following comparators:

=
: Is equal to.

<>
: Is not equal to.

<
: Is less than.

>
: Is greater than.

<=
: Is less than or equal to.

>=
: Is greater than or equal to.

is
: When you search for `null` values:

  is null
  : Indicates a null (or no) value.

  is not null
  : Indicates a valid value that is not null.

Conjunctions
You can also use conjunctions.

AND
: Tie together multiple filter criteria.

OR
: Meet at least one of multiple filter criteria.

Helpers
You can also use helpers.

NOW()
: Get the current TIMESTAMP.

DAYS/MONTHS/YEARS
: Compares TIMESTAMP/DATE values.

Wildcards
You can also use a wildcard.

%
: You can use a wildcard like % with the keyword LIKE to form a wildcard search.

Note: Both single quotation marks (') and backslashes (\) need to be escaped with a leading backslash. For example, if the search statement is: `mytag in ('first', 'can't', 'with\slash')`, the following convention needs to be followed to escape it:
`mytag in ('first', 'can\'t', 'with\\slash')`.

Examples of search filters

This section provides a list of examples for search filters.

Note: You must wrap string values in single quotation marks but you cannot wrap numeric values in single quotes.

Owner=**'bob'**
    # All files owned by **'bob'**.
Fileset=**'bobs project'**
    # All files in the file set **bobs_project**.
Filetype = **'pdf'** AND size > 500000
    # All PDF files that are larger than 500000 bytes.
Filetype is **('txt', 'pdf', 'doc')**
    # All files of type TXT, PDF, or DOC.
Atime < (NOW() - 180 DAYS)
    # All files not accessed in the last 180 days.
Filesystem = **'big_fs '** AND owner <> **'root'**
    # All files in the **big_fs** filesystem that are not owned by root.
collection = **'proj_xylem'**
    Search for all records that are tagged with the user-defined tag **'Project'** set to **'proj_xylem'**.
collection <> ''
    Search for all records that have a collection that is assigned.
filename LIKE **'the_quick_brown_%'**
    Returns all records for which the file name begins with **"the_quick_brown_".**
department= **'department_xylem'**
    Search for all records that are tagged with the user-defined tag **'Department'** set to **'proj_xylem'**.
custom_tag is null
    # All files for which **custom_tag** is not set to any value.

Search results table

The search results table displays information about the records that met the search criteria.

By default, certain columns are shown, and others are hidden. Click  next to the search bar to customize the column view.

Refine search results

After a set of search results is returned, you can further refine the data by using the following options:

- Under Grouped By, click the relevant criteria to group the search results.
- Select a record row and click View individual record on the header to view the record details.
- Click Back to Query builder and select the size of search results you want to view from Limit results.
- Click  and select the Used capacity column to view a comparative study of the file size that is reported and the amount of space that is being used in the storage.

You can use a combination of any or all of the filtering criteria. The search results are immediately refined based on the criteria you select.

Sort search results

You can sort search results by column.

When you click the column header, you can sort the results in ascending order. When you click the column a second time, you can sort the column in descending order. The time it takes to sort depends on the size of the result set.
Note: Sorting by a second column loses the order of the data in the first column. A combination sort view is not supported.

Tag search results manually

After a filtered set of records is displayed on the search pane, you can select all or some of those documents and apply organizational tags to them.

For example, if a drill-down search result identifies all of the records for a particular project, you can assign a specific tag to it. Click Add Tags and specify that an organizational tag called 'Project' is set to the name of the project that is represented by the filtered set. The tag application runs as a background task and you get a notification when the processing is complete.

You can apply more than one tag at the same time.

Managing applications

An application is a program that interfaces with IBM Data Cataloging and can access the source storage. There are many use cases for application, including data content inspection for enriching metadata, data movement or migration, data scrubbing or sanitation, and more. Data is identified by IBM Data Cataloging by policy filter and passed to the application as pointers through a messaging queue. Then, the application performs whatever work is appropriate on the source data and returns a completion status back to IBM Data Cataloging, which might or might not include enriched metadata for the records. If it does include enriched metadata, IBM Data Cataloging catalogs that metadata and makes it immediately searchable.

Permissions

Data Administrator
: Create (register), update, delete (unregister), and view the applications.

Data User
: View the applications created by a Data Administrator.

Security Administrator
: Cannot create, modify, view, or delete any application.

Management

Applications might be viewed and deleted by navigating to Metadata > Applications. You can define an application when you are creating a new **DEEP-INSPECT** policy. In addition, you can add Parameters for an application during the process of creating a **DEEP-INSPECT** policy. For more information, see Adding deep-inspection policy parameters .

Figure 1. Applications table



The Applications table displays the following information:

Application
: The name of the application.

Parameters
: The parameters that were assigned to the application when the policy was created.

Action ID
: **AUTOTAG** or **deepinspect** - the policy type that the application is assigned to.

For more information, see Application management by using APIs.

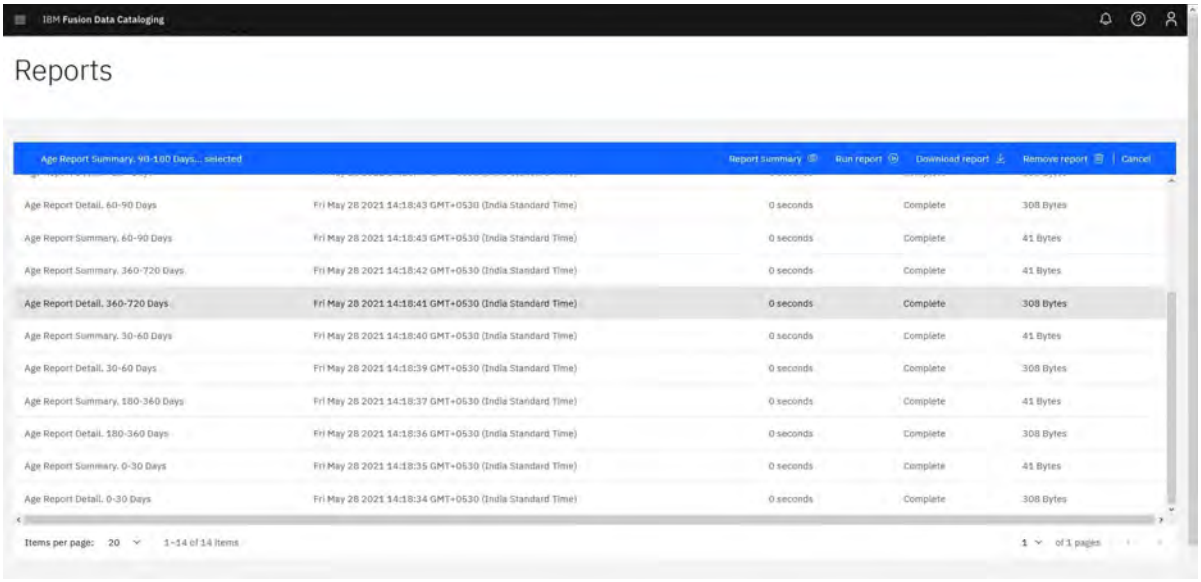
Reports

Reports can be generated upon applying tags to a set of data.

Procedure

1. Reports can be generated by using the following methods:
   - Discover data by performing a Search in IBM Data Cataloging. The search results provide an option to Generate Reports. For more information, see Searching for details.
   - Use the Graphical User Interface (GUI)Graphical User Interface (GUI) to automatically run the reports during deployment.
2. Click menu and go to Search > Reports .

   Figure 1. Reports table

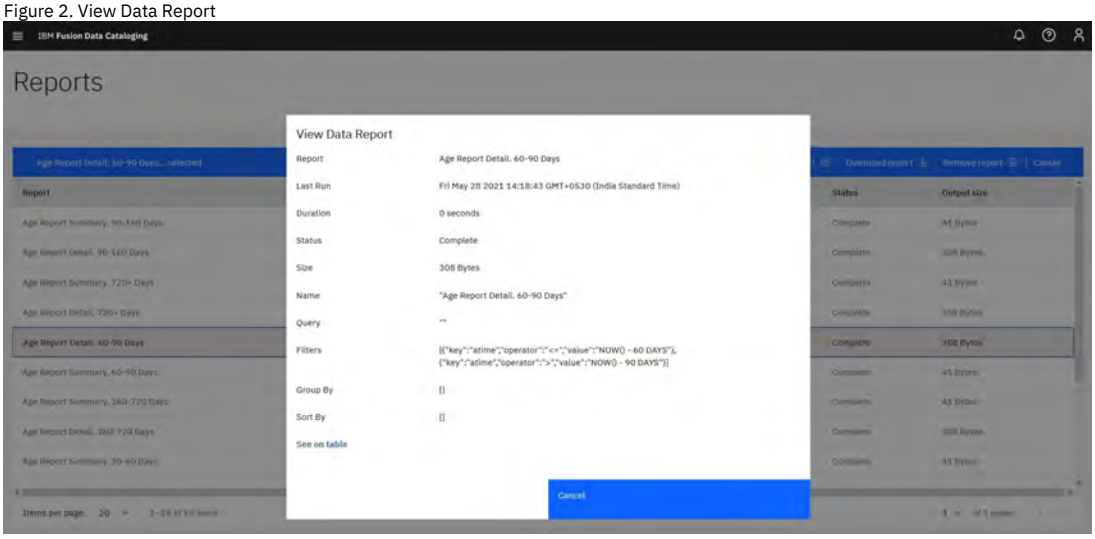


3. The following actions can be completed in a table:

   Report Summary

   a. To view a report, select a report and click Report summary. The report's statistics are displayed in a View Data Report dialog.

   Figure 2. View Data Report



   b. Click See on table to view all the records of a report. The Query Results window displays the results of the search.

   Download report

   Click Download report to open a report with a text editor, or to save the report to local storage.

   Run report

   To run a report, select a report and click Run report.

   Remove report

   To remove a report, select a report and click Remove report.

Creating a IBM Data Cataloging application for metadata-based policies

Provides information about how to create IBM Data Cataloging application for metadata-based policies.

By default, a IBM Data Cataloging application creates a privileged container prepared to process a policy that requires both metadata and content inspection, which oftentimes mounts a data source to fetch the required files. To modify this behavior and create non-root applications for metadata-only policies, we can add the non-root annotation to the resource.

For example:

```
apiVersion: spectrum-discover.ibm.com/v1alpha1
kind: SpectrumDiscoverApplication
metadata:
  annotations:
    nonroot: 'true'
  name: non-root-app
  namespace: ibm-data-cataloging
spec:
  application_name: nonroot
```

```
log_level: DEBUG
replicas: 1
repo_name: <PRIVATE_REGISTRY>/metadata-example-application
tag: v1.0.0
```

Graceful shutdown

Provides detailed instructions to put IBM Data Cataloging in idle state on an OpenShift® environment.

About this task

The procedure might be helpful in situations where IBM Data Cataloging is behaving unexpectedly or there is a maintenance window of not more than one hour.

Procedure

1. Stop or wait for all running scans and policies to stop. Do not skip this step until everything looks stopped.
2. Do `oc` login to your OpenShift Container Platform cluster.
3. Do `oc project <project>` to select namespace where IBM Data Cataloging service is installed.
4. Stop all pods and set replicas to zero for the following deployments. and respect the following order:
   Note: Follow the suggested sequence.

   a. Consumers

      i. Run the following command to get the list of the deployments and filter by consumer.

      ```
      $oc get deployments |grep consumer
      ```

      Example output:

      ```
      isd-consumer-ceph-le                  10/10   10           10          6d20h
      isd-consumer-cos-le                   10/10   10           10          6d20h
      isd-consumer-cos-scan                 10/10   10           10          6d20h
      isd-consumer-file-scan                10/10   10           10          6d20h
      isd-consumer-protect-scan             10/10   10           10          6d20h
      isd-consumer-scale-le                 10/10   10           10          6d20h
      isd-consumer-scale-scan               10/10   10           10          6d20h
      ```

      ii. Run the following to command to scale down replicas to 0 for every consumer deployment.

      ```
      $oc scale --replicas=0 deployment <deployment_name>
      ```

      Example output:

      ```
      $oc scale --replicas=0 deployment isd-consumer-file-scan
      deployment.apps/ isd-consumer-file-scan scaled
      ```

      iii. Ensure you scale down replicas to 0 for every consumer deployment.
      Example output:

      ```
      $oc get deployments|grep consumer-cos-scan
      isd-consumer-file-scan                  0/0     0            0           6d20h
      ```

   b. Producers

      i. Run the following command to get the list of the deployments and filter by producer.

      ```
      $oc get deployments |grep producer
      ```

      Example output:

      ```
      isd-producer-ceph-le                  1/1     1            1           6d20h
      isd-producer-cos-le                   1/1     1            1           6d20h
      isd-producer-cos-scan                 1/1     1            1           6d20h
      isd-producer-file-scan                1/1     1            1           6d20h
      isd-producer-protect-scan             1/1     1            1           6d20h
      isd-producer-scale-le                 1/1     1            1           6d20h
      isd-producer-scale-scan               1/1     1            1           6d20h
      ```

      ii. Run the following to command to scale down replicas to 0 for every producer deployment.
      Important: Ensure that you scale down replicas to 0 for every producer deployment.

      ```
      $oc scale --replicas=0 deployment <deployment_name>
      ```

      Example output:

      ```
      $oc scale --replicas=0 deployment isd-producer-file-scan
      deployment.apps/ isd-producer-file-scan scaled
      ```

      iii. Ensure that you have scale down replicas to 0 for every producer deployment.
      Example output:

      ```
      $oc get deployments|grep producer-cos-scan
      isd-producer-cos-scan                   0/0     0            0           6d20h
      ```

   c. Connection manager

      i. Run the following command to get the list of the statefulsets and filter by connection manager.

      ```
      $oc get statefulsets |grep connmgr
      ```

Example output:

```
$oc get statefulsets |grep connmgr
connmgr                    0/0     0            0           6d20h
```

ii. Run the following to command to scale down replicas to 0 for every connection manager statefulset.
Important: Ensure that you have scale down replicas to 0 for every connection manager statefulset.

```
$oc scale --replicas=0 statefulset
```

Example output:

```
$oc scale --replicas=0 statefulset connmgr
statefulset.apps/ connmgr scaled
```

iii. Ensure that you have scale down replicas to 0 for every connection manager statefulset.
Example output:

```
$oc get statefulsets|grep connmgr
connmgr                    0/0     0            0           6d20h
```

d. Check whether db2whrest stopped inserting data properly
Ensure that no database activity is present before stopping db2whrest. It is necessary to check that specific csv files created are empty, which means there are no pending jobs being processed.

i. Run the following command to go to db2whrest.

```
oc rsh deployment/isd-db2whrest
```

Now, list all CSV files related to merging that are empty.

For example:

```
$for file in $(find /gpfs/gpfs0/db2wh/home/bluadmin -name *LOAD*.csv); do file $file; done |grep data
/gpfs/gpfs0/db2wh/home/bluadmin/ACOG_LOAD4.csv: data
/gpfs/gpfs0/db2wh/home/bluadmin/ACOG_LOAD1.csv: data
$
```

ii. Wait for a few seconds and try again until you get an empty list.

e. DB2WHRest

i. Run the following command to get the list of the deployments and filter by db2whrest.

```
$oc get deployments |grep db2whrest
```

Example output:

```
$oc get deployments |grep db2whrest
isd-db2whrest                  1/1     1            1           6d20h
```

ii. Run the following command to scale down replicas to 0 for every db2whrest deployment.

```
$oc scale --replicas=0 deployment <deployment_name>
```

Example output:

```
$oc scale --replicas=0 deployment db2whrest
deployment.apps/ db2whrest scaled
```

iii. Ensure that all scale down replicas are 0 for every db2whrest deployment.
Example output:

```
$oc get deployments|grep db2whrest
isd-db2whrest                  0/0     0            0           6d20h
```

5. Stop Db2 instance. For procedure, see Stopping and starting a Db2 instance.
6. After checking that everything is down, IBM Data Cataloging is successfully in an idle state. It is recommended to be in this state for a short period, not more than one hour. Warning messages on other IBM Data Cataloging pods are expected.

- **Flushing Kafka topics**
  You must have kafka to avoid issues related to restore the specified flushed out topic.
- **Returning IBM Data Cataloging to a running state**
  Provides detailed instructions to put IBM Data Cataloging in running state on an OpenShift environment.

Flushing Kafka topics

You must have kafka to avoid issues related to restore the specified flushed out topic.

Procedure

1. Run the following command to get Kafka topics.

```
oc get kafkatopic |grep file-scan-topic
```

Example output:

```
File-scan-topic                isd     10      1       True
```

2. Run the following command to back up Kafka topic.

   ```
   oc get kafkatopic file-scan-topic -o yaml
   ```

   Example output:

   ```
   apiVersion: kafka.strimzi.io/v1beta2
   kind: KafkaTopic
   metadata:
     resourceVersion: '15457887'
     name: file-scan-topic
     namespace: ibm-data-cataloging
     ownerReferences:
       - apiVersion: spectrum-discover.ibm.com/v1alpha1
         kind: SpectrumDiscover
         name: discover-instance
         uid: 52fd9b6c-0823-4f88-b43d-525dd6992b1b
     labels:
       strimzi.io/cluster: isd-ssl
   spec:
     partitions: 10
     replicas: 1
   ```

3. Run the following command to delete the Kafka topic to flush the data.

   ```
   oc delete kafkatopic file-scan-topic
   ```

4. Run the following command to apply the backed-up YAML file on the cluster to recreate the topic until it deletes the Kafka topic.

   ```
   oc apply -f <topic.yaml>
   ```

   Example output:

   ```
   apiVersion: kafka.strimzi.io/v1beta2
   kind: KafkaTopic
   metadata:
     resourceVersion: '15457887'
     name: file-scan-topic
     namespace: ibm-data-cataloging
     ownerReferences:
       - apiVersion: spectrum-discover.ibm.com/v1alpha1
         kind: SpectrumDiscover
         name: discover-instance
         uid: 52fd9b6c-0823-4f88-b43d-525dd6992b1b
     labels:
       strimzi.io/cluster: isd-ssl
   spec:
     partitions: 10
     replicas: 1
   ```

Returning IBM Data Cataloging to a running state

Provides detailed instructions to put IBM Data Cataloging in running state on an OpenShift® environment.

About this task

Follow the steps in a specific sequence to put IBM Data Cataloging in to a running state from idle.

Procedure

1. Start Db2 instance. For procedure, see Stopping and starting a Db2 instance.
2. It is needed to come back to a running state and ensure that you make replicas back to the previous state in the following order.
   Note: It is important to scale up smoothly for those deployments that have two or more replicas and wait for them to be at the ready stage to scale up the next two replicas.

   a. DB2WHRest

      i. Run the following command to get the list of the deployments and filter by db2whrest.

      ```
      $oc get deployments |grep db2whrest
      ```

      Example output:

      ```
      $oc get deployments |grep db2whrest
      isd-db2whrest                    0/0     0            0           6d20h
      ```

      ii. Run the following to command to scale up replicas to 1 for every db2whrest deployment.

      ```
      $oc scale --replicas=1 deployment <deployment_name>
      ```

      Example output:

      ```
      $oc scale --replicas=1 deployment db2whrest
      deployment.apps/ db2whrest scaled
      ```

      iii. Ensure that you have scale up replicas to 1 for every db2whrest deployment.
      Example output:

```
$oc get deployments |grep db2whrest
isd-db2whrest                    1/1     1            1           6d20h
```

b. Connection manager

i. Run the following command to get the list of the deployments and filter by connection manager.

```
$oc get deployments |grep connmgr
```

Example output:

```
$oc get deployments |grep connmgr
isd-connmgr                    0/0     0            0           6d20h
```

ii. Run the following to command to scale down replicas to 1 for every connection manager deployment.

```
$oc scale --replicas=1 deployment <deployment_name>
```

Example output:

```
$oc scale --replicas=1 deployment connmgr
deployment.apps/ connmgr scaled
```

iii. Ensure that you have scale up replicas to 1 for every connection manager deployment.
Example output:

```
$oc get deployments |grep connmgr
isd-connmgr                    1/1     1            1           6d20h
```

c. Producers

i. Run the following command to get the list of the deployments and filter by producer.

```
$oc get deployments |grep producer
```

Example output:

```
isd-producer-ceph-le                    0/0     0            0           6d20h
isd-producer-cos-le                     0/0     0            0           6d20h
isd-producer-cos-scan                   0/0     0            0           6d20h
isd-producer-file-scan                  0/0     0            0           6d20h
isd-producer-protect-scan               0/0     0            0           6d20h
isd-producer-scale-le                   0/0     0            0           6d20h
isd-producer-scale-scan                 0/0     0            0           6d20h
```

ii. Run the following to command to scale up replicas to 1 for every producer deployment.

```
$oc scale --replicas=1 deployment <deployment_name>
```

Example output:

```
$oc scale --replicas=1 deployment isd-producer-file-scan
deployment.apps/ isd-producer-file-scan scaled
```

iii. Tip: It is recommended to scale up replicas for two producers and continue with the next two only when they are ready, up-to-date, and available. The object of this is to give OpenShift time to handle these requests properly.
Ensure that you have scale up replicas to 1 for every producer deployment.
Example output:

```
$oc get deployments|grep producer-cos-scan
isd-producer-cos-scan                   1/1     1            1           6d20h
```

d. Consumers

i. Run the following command to get the list of the deployments and filter by consumer.

```
$oc get deployments |grep consumer
```

Example output:

```
isd-consumer-ceph-le                    0/0     0            0           2d
isd-consumer-cos-le                     0/0     0            0           2d
isd-consumer-cos-scan                   0/0     0            0           2d
isd-consumer-file-scan                  0/0     0            0           2d
isd-consumer-protect-scan               0/0     0            0           2d
isd-consumer-scale-le                   0/0     0            0           2d
isd-consumer-scale-scan                 0/0     0            0           2d
```

ii. Run the following to command to scale up replicas to 2 for every consumer deployment.

```
$oc scale -replicas=2 deployment isd-consumer-file-scan
```

Example output:

```
$oc scale -replicas=2 deployment isd-consumer-file-scan
deployment.apps/isd-consumer-file-scan scaled
```

iii. Tip: It is recommended to scale up replicas for two producers and continue with the next two only when they are ready, up-to-date, and available. The object of this is to give OpenShift time to handle these requests properly.
Ensure that you have scaled up replicas to 10 for every consumer deployment, but you must scale up two at a time.
Example output:

```
$oc get deployments|grep isd-consumer-file-scan
isd-consumer-file-scan          2/2     2       2       2d
```

iv. Some pods might face issues when scaling up the replicas back to 10 in a row. If that happens, then you need to wait for a few minutes, but if it is back to the Running or Ready (1/1) state for a long time, then you need to check for failing pods and delete them. OpenShift will create them

again after you delete them.

1. Run the following to get the list of pods that are not running.

   ```
   $oc get pods | grep -v Running
   ```

   Example output:

   ```
   isd-consumer-file-scan-6d6bbf8487-2asd3                          0/1     Pending   0               2h
   isd-consumer-file-scan-6d6bbf8487-ljkws                           0/1    Pending   0               2h
   isd-consumer-file-scan-6d6bbf8487-m96hc                          0/1     Pending   0               2h
   isd-consumer-file-scan-6d6bbf8487-w551r                           0/1    Pending   0               2h
   isd-consumer-file-scan-6d6bbf8487-z8hqd                           0/1    Pending   0               2h
   ```

2. Run the following command to delete pods that are not running.

   ```
   $oc delete pod isd-consumer-file-scan-6d6bbf8487-ljkws
   ```

   Example output:

   ```
   $oc delete pod isd-consumer-file-scan-6d6bbf8487-ljkws
   pod "isd-consumer-file-scan-6d6bbf8487-ljkws" deleted

   $ oc get pods isd-consumer-file-scan-6d6bbf8487-ljkws
   NAME                                        READY    STATUS         RESTARTS    AGE
   isd-consumer-file-scan-6d6bbf8487-ljkws    1/1      Terminating    0           2h
   ```

3. OpenShift will create them again after you delete them.
   For example:

   ```
   $oc get pods|grep consumer-file-scan
   isd-consumer-file-scan-6d6bbf8487-4lx8r                            0/1     Running     0
   64s

   #Check until having 1/1 Running state

   $oc get pods|grep consumer-file-scan
   isd-consumer-file-scan-6d6bbf8487-4lx8r                            1/1     Running     0
   2m39s
   ```

REST API for IBM Data Cataloging

REST API reference documentation provides endpoint specifications, request and response formats, and authentication details for programmatic access to IBM Spectrum Discover services.

- **IBM Data Cataloging REST APIs**
  The Data Cataloging REST APIs are REST-style APIs that provide interoperability between a client and server over a network. These APIs allow authenticated users to perform management tasks.
- **Endpoints for working with a Db2 Warehouse**
  The Db2® Warehouse API provides endpoints for working with a Db2 Warehouse.
- **Endpoints for working with policy management**
  With the policy management API service, you can create, list, update, and delete the policies.
- **Endpoints for working with connection management**
  With the connection management API service you can create, update, delete, and get information about data source connection entries in the connection table.
- **Application management by using APIs**
  The application management APIs provide options to get the details of the available applications, install or upgrade an application, and delete an application. You must have the Admin role to access the application management endpoints.
- **RBAC management by using APIs**
  The IBM Data Cataloging resource-based access control (RBAC) is a REST API service that enables role-based access to the IBM Data Cataloging services. This service uses OpenStack Keystone as a backend for providing Identity and Access Management (IAM) across multiple domains that are attached to the IBM Data Cataloging.

IBM Data Cataloging REST APIs

The Data Cataloging REST APIs are REST-style APIs that provide interoperability between a client and server over a network. These APIs allow authenticated users to perform management tasks.

The following list shows the significant features of REST-style APIs:

- REST-style APIs are resource-based.
- REST-style APIs are stateless.
- REST-style APIs are client or server.
- REST-style APIs are cacheable.
- REST-style APIs are a layered system.

A representational state transfer (REST) system is a resource-based service system in which requests are made to the resource's universal resources identifier (URI). These requests start a response from the resource in the JSON or CSV format.

The operations that you can perform on the resources or a resource element are directed by the HTTP methods such as GET, POST, PUT, and DELETE and in some cases by the parameters of the HTTPS request. The following list provides the meanings of the basic HTTP methods that are used in the requests:

GET
: Reads a specific resource or a collection of resources and provides the details as the response.

PUT
: Updates a specific resource or a collection of resources.

DELETE
: Removes or deletes a specific resource.

POST
: Creates a resource.

- **API endpoints**
  The Data Cataloging REST APIs include several API services for monitoring data and performing other management tasks.
- **Asynchronous endpoints**
  Some REST API endpoints run asynchronously.
- **Status codes**
  Each API request that is sent to the server returns a response that includes an HTTP status code and any requested information.
- **Authentication process**
  The REST API services require token-based authentication rather than authentication with a user ID and password.

API endpoints

The Data Cataloging REST APIs include several API services for monitoring data and performing other management tasks.

IBM Data Cataloging provides the following REST APIs:

- An API for working with a DB2® warehouse
- A policy management API and an autotags API
- A connection management API
- An application management API
- A resource-based access control (RBAC) API

The endpoints of each API have a characteristic basic syntax. In the following code blocks, **data_cataloging_host** is the host name or IP address of the API server:

- Endpoints for the Db2® Warehouse API:

  ```
  https://<data_cataloging_host>/db2whrest/v1/<endpoint_ID>
  ```

  For example:

  ```
  curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ sql_query_async?<sql>
  ```

- Endpoints for the policy management API:

  ```
  https://<data_cataloging_host>/policyengine/v1/policies/<endpoint_ID>
  ```

  For example:

  ```
  curl -k -H 'Authorization: Bearer <token>' "https://<data_cataloging_host>/policyengine/v1/ policies/<policy_ID>
  ```

- Endpoints for the resource-based access control (RBAC) API:

  ```
  https://<data_cataloging_host>/auth/v1/<endpoint_ID>
  ```

  For example:

  ```
  curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users
  ```

Asynchronous endpoints

Some REST API endpoints run asynchronously.

This behavior is helpful if the action that is performed has a long waiting period or has a negative impact on system performance. Some endpoints are explicitly asynchronous, such as the **/db2whrest/v1/sql_query_async?<sql>: GET** endpoint. Other endpoints are initially synchronous but are automatically converted by the system to asynchronous operations if they take more than 10 seconds to complete.

When you start an endpoint that results in an asynchronous operation, you must poll the asynchronous operation until it is complete. Follow these steps:

1. Find the *location URI* of the asynchronous operation. In response to the request that starts the endpoint, the system returns a synchronous response that indicates that asynchronous operation is accepted for processing but is not yet complete. This response includes a status code of **202 ACCEPTED** and a response header that contains the location of the asynchronous operation. In the following example, the status code appears at line 2 and the location URI appears in line 5:

   ```
   * HTTP 1.0, assume close after body
   < HTTP/1.0 202 ACCEPTED
   < Content-Type: text/html; charset=utf-8
   < Content-Length: 28
   < Location: https://labsrv04/db2whrest/v1/task_status/9c0db090-bd33-41c5-969f-a1603ddf49ab
   < Server: Werkzeug/0.14.1 Python/3.4.5
   < Date: Thu, 15 Feb 2018 20:37:10 GMT
   ```

2. Run the **/db2whrest/v1/task_status/<*task_id*>: GET** endpoint on the location URI to get the status of the asynchronous operation. The following example uses the location URI from the example in Step 1:

   ```
   curl -k -H 'Authorization: Bearer <token>' https://labsrv04/db2whrest/v1/task_status/9c0db090-bd33-41c5-969f-a1603ddf49ab
   -X GET
   ```

3. If the status code that is returned from the task_status endpoint in Step 2 is **303**, then the asynchronous operation is not completed. Repeat Step 2.

4. If the status code of the task_status endpoint in Step 2 is `200`, then the asynchronous operation completes and the response header contains the location URI of the results. Run the `/db2whrest/v1/task_status/<task_id>: GET` endpoint on this location URI to get the results.

For more information see the topic /db2whrest/v1/task_status/<task_id>: GET and <task_id>/peek.

Status codes

Each API request that is sent to the server returns a response that includes an HTTP status code and any requested information.

The following are some of the common HTTP status codes:

200 OK
: The endpoint operation was successful.

201 Created
: The endpoint operation was successful and resulted in the creation of a resource.

202 Accepted
: The request is accepted for processing, but the processing is not yet completed. Asynchronous endpoints return this status code in the response to the original request.

204 No content
: The endpoint operation was successful, but no content is returned in the response.

303 [interim response status]
: The endpoint operation is in progress. Asynchronous endpoints return this status code in response to a request for status.

The following are some common HTTP status error codes:

400 Bad Request
403 Forbidden
404 Not Found
500 Internal Server Error

Along with the HTTP status codes, the Db2® Warehouse API service also returns an error reason in the `reason` field of a JSON dictionary. In the following example, a request to add a record to a database is rejected with the status error code 400 "BAD REQUEST". The reason code contains more information, including the following error message:

```
curl -k -v -H 'Authorization: Bearer <token>'
https://data_cataloging_host/db2whrest/v1/sql_query -X POST -d"select *
from labsrv04 where ownerzzz='root'"
<HTTP/1.1 400 BAD REQUEST
{"reason": "('42S22', '[42S22] [IBM][CLI Driver][DB2/LINUXX8664] SQL0206N  \"OWNERZZZ\" is not
valid in the context where it is used.  SQLSTATE=42703\\n')", "status": "error"}
```

Authentication process

The REST API services require token-based authentication rather than authentication with a user ID and password.

Users who need to access the IBM Data Cataloging APIs need to get an authentication token first by using their username and password. Then, use that token to get authenticated to the IBM Data Cataloging system to perform various operations by using APIs.

Authentication is achieved in IBM Data Cataloging through the following steps:

1. The administrator registers an enterprise domain, which can be either Lightweight Directory Access Protocol (LDAP) or Cloud Storage Object, with the authentication service. Registering the enterprise domain with the authentication service includes the following steps:
    a. The administrator gets an authentication token by using the credentials.
    When the administrator (who is in the LDAP domain) logs in to IBM Data Cataloging, the credentials are passed from IBM Data Cataloging to LDAP for authentication. Then, the administrator gets an authentication token by using the credentials.
    Note: Users from an external LDAP or Cloud Storage Object domain, need to include domain name in the user name as `"<domain>/<user>"` to get an authentication token by using the REST APIs.
    b. Register the domain by using the obtained authentication token.
    IBM Data Cataloging integrates with both LDAP users and Cloud Storage Object users. Administrators can add the Cloud Storage Object domain to IBM Data Cataloging and the users are imported into IBM Data Cataloging where the administrator can add the users to the appropriate collections.

    In this case, the Cloud Storage Object users can either use their user name and password or the Cloud Storage Object native Amazon Simple Storage Service (Amazon S3) access key and secret key pair to get authenticated with the IBM Data Cataloging authentication service to get an authentication token.

    Like the users from LDAP domain, the Cloud Storage Object domain users can use this authentication token to access the IBM Data Cataloging services and their scope is restricted to the projects to which they can access.
2. The administrator adds collections to the authentication service and adds users to these collections by assigning them appropriate roles. The records that the users can see or apply policies to are restricted according to the collections to which they have access.
3. The user requests for an authentication token by using their user name and password.
The IBM Data Cataloging RESTful service like DB2WH REST, policy engine, tags, applications, and the various authentication service endpoints require a bearer auth-token to be passed to it in the authorization header.

    The authentication token needs to be obtained by using the authentication token service endpoint, which is the endpoint for a user to log in with user name and password credentials, by using HTTP basic authentication. This token then can be used for authorization across various service endpoints. After a user receives the authentication token, it is valid for 1 hour. Using this token, a user can use various IBM Data Cataloging services.

4. Users can access the IBM Data Cataloging services by using the authentication token.

For more information, see /auth/v1/token: GET.

Endpoints for working with a Db2 Warehouse

The Db2® Warehouse API provides endpoints for working with a Db2 Warehouse.

- **/db2whrest/v1/search: POST**
  Searches a database for data.
- **/db2whrest/v1/report: POST**
  Creates a curation report definition and runs it immediately.
- **/db2whrest/v1/report: GET**
  Gets information on all the curation reports that are available in the system.
- **/db2whrest/v1/report/<report_id>: GET**
  Gets information about a single curation report.
- **/db2whrest/v1/report/<report_id>/download: GET**
  Downloads the output file of the specified curation report.
- **/db2whrest/v1/bucket: GET**
  Gets information on all buckets that are supported.
- **/db2whrest/v1/buckets/<bucket>: GET**
  Gets information on a specific bucket that is supported.
- **/db2whrest/v1/report/<report_id>: PUT**
  Runs the specified curation report. The old output file is replaced by the new one.
- **/db2whrest/v1/buckets/<bucket>: PUT**
  Modifies the bucket details.
- **/db2whrest/v1/report/<report_id>: DELETE**
  Deletes the specified curation report definition and its output file.
- **/db2whrest/v1/sql_query?<sql>: GET**
  Retrieves data from a database with an SQL query in a GET command. You can use any standard SQL query. You must include the SQL query as a parameter of the URL.
- **/db2whrest/v1/sql_query: POST**
  Gets data from a database with an SQL query in a POST command. You can use any standard SQL query. You must include the SQL query in the message body of the POST command.
- **/db2whrest/v1/sql_query_async?<sql>: GET**
  Gets data from a database asynchronously with an SQL query in a GET command.
- **/db2whrest/v1/sql_query_async: POST**
  Gets data from a database asynchronously with an SQL query in a POST command. You can use any standard SQL query. You must include the SQL query in the message body of the POST command.
- **/db2whrest/v1/task_status/<task_id>: GET and <task_id>/peek**
  Gets the status of the specified asynchronous task.
- **/db2whrest/v1/summary_tables -X GET**
  Gets information on all the summary tables that are available in the system.
- **/db2whrest/v1/summary_tables/<table> -X GET**
  Gets information for a particular summary table available in the system.
- **/db2whrest/v1/summary_tables/<table> -X PUT**
  Change the summary table configuration for a particular summary table available in the system.
- **/db2whrest/v1/summary_tables/<table>/<action> -X PUT**
  Triggers a start or scheduled start action for a particular summary table.
- **/db2whrest/v1/bulk_add_tags/docs: POST**
  Searches a database for data.

/db2whrest/v1/search: POST

Searches a database for data.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	√[1]	√[1]	X	X

[1]The search is restricted to documents that are tagged with collections to which the user ID has a datauser role assigned.

The **search** endpoint has the following format:

```
/db2whrest/v1/search -H 'Authorization: Bearer <token>' -X POST -d@<data.json>
-H "Content-Type: application/json"
```

The endpoint accepts requests in JSON format and returns a response in a self-describing data set. It also returns the query time and number of elements in the result set. If an endpoint operation takes more than 10 seconds to complete it is converted to an asynchronous operation. For more information on asynchronous endpoint operations, see Asynchronous endpoints.
You can modify a **search** endpoint with the following fields. An example follows this list:

query
: Specifies a search query string.

filters
: Specifies an array of dictionaries that filter the results of the query. Each dictionary must contain the following three fields:

  key

The name of the field or column to be returned.

operator
: One of the following operators: `=, >, <, <>, <=, >=, in, like`.

value
: The value of the field or column to filter on.

group_by
: Specifies a list of fields to be used to summarize search results. If the group_by field is not specified, the search returns record-level information.
Note: The combination `"group_by": ["filename"]` causes the query to be applied to the duplicate file table. All other group_by combinations cause the query to be applied to the summary tables.

sort_by
: Specifies an array of dictionary objects. Each dictionary object must specify a field or column name to be sorted on and a sort direction. Valid sort directions are asc and desc.

limit
: Specifies the maximum number of rows that can be returned by the response.

The following example illustrates how to specify these fields:

```
{
    "query": "platform='Spectrum Scale'",
    "filters": [
      {
        "key": "project",
        "operator": "is",
        "value": "null"
      }
    ],
    "group_by": ["Filesystem","Owner","Site"],
    "sort_by": [{"Filesystem": "asc"},{"Owner": "asc"}],
    "limit": 100000
}
```

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/search -X POST -d@search.json -H
"Content-Type: application/json"
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

1. The following example shows how to define search parameters and format the data that is returned:
   a. Step 1: Define the search parameters in a file named search.json:

   ```
   {
       "query": "platform='Spectrum Scale'",
       "filters": [
         {
           "key": "project",
           "operator": "is",
           "value": "null"
         }
       ],
       "group_by": ["Filesystem","Owner","Site"],
       "sort_by": [{"Filesystem": "asc"},{"Owner": "asc"}],
       "limit": 100000
   }
   ```

   b. Step 2: Submit the request:

   ```
   curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ search -X POST -d@search.json
   -H "Content-Type: application/json
   ```

   The following response is returned:

   ```
   {
   "facet_tree": {
   "OWNER": "[{\"owner\":\"nobody\",\"count\":8},{\"owner\":\"benjamin\",\"count\":2},
   {\"owner\":\"_NULL_\",\"count\":989},{\"owner\":\"borgato\",\"count\":2},
   {\"owner\":\"boston\", \"count\":4554},{\"owner\":\"root\",\"count\":366104932},
   {\"owner\":\"beale\",\"count\":203545},
   {\"owner\":\"baldwin\",\"count\":2},{\"owner\":\"behr\",\"count\":785144},
   {\"owner\":\"babcock\", \"count\":9082943},{\"owner\":\"broadwood\",\"count\":375}]",
   "PLATFORM": "[{\"platform\":\"Spectrum Scale\",\"count\":376182496}]",
   "FILESYSTEM": "[{\"filesystem\":\"fs11-1m-me1\",\"count\":185493076},
   {\"filesystem\":\"filesys1\",\"count\":10077057},{\"filesystem\":\"fs10-1m-me1\",
   \"count\":180612363}]",
   "CLASSIFICATION": "[{\"classification\":null,\"count\":376182496}]",
   "DEPARTMENT": "[{\"department\":null,\"count\":376182496}]",
   "CLUSTER": "[{\"cluster\":\"host.ibm.com\",\"count\":366105439},{\"cluster\":
   ```

```
\"filesys1.university.edu\",\"count\":10077057}]",
"TIER": "[{\"tier\":\"system\",\"count\":376182496}]",
"ARCHIVE": "[{\"archive\":null,\"count\":376182496}]",
"SITE": "[{\"site\":\"host\",\"count\":366105439},{\"site\":\"\",\"count\":10077057}]",
"PROJECT": "[{\"project\":null,\"count\":376182496}]"},
"query_time_secs": 1.174846,
"rows": "[{\"filesystem\":\"fs10-1m-me1\",\"owner\":\"_NULL_\",\"site\":\"host\",
\"count\":468,
\"sum\":52933001066},{\"filesystem\":\"fs10-1m-me1\",\"owner\":\"nobody\",\"site\":
\"host\", \"count\":4,\"sum\":20976},{\"filesystem\":\"fs10-1m-me1\",\"owner\":\"root\",
\"site\":\"host\",\"count\":180611891,\"sum\":2002705351263},{\"filesystem\":
\"fs11-1m-me1\",\"owner\":\"_NULL_\", \"site\":\"host\",\"count\":521,\"sum\":58473947666},
{\"filesystem\":\"fs11-1m-me1\",\"owner\":\"nobody\",\"site\":\"host\",\"count\":4,\"sum\":20976},{\"filesystem\":\"fs11-
1m-me1\",
\"owner\":\"root\",\"site\":\"host\",\"count\":185492551,\"sum\":2174830065250},
{\"filesystem\":\"filesys1\",\"owner\":\"baldwin\",\"site\":\"\",\"count\":2,
\"sum\":16388},{\"filesystem\":\"filesys1\",\"owner\":\"behr\",\"site\":\"\",
\"count\":785144,\"sum\":5000771899189}, {\"filesystem\":\"filesys1\",
\"owner\":\"boston\",\"site\":\"\",\"count\":4554,\"sum\":57670240959030},
{\"filesystem\":\"filesys1\",\"owner\":\"beale\",\"site\":\"\",\"count\":203545,
\"sum\":69505800364825},{\"filesystem\":\"filesys1\",\"owner\":\"root\",\"site\":
\"\",\"count\":490,\"sum\":265209686947},{\"filesystem\":\"filesys1\",\"owner\":
\"broadwood\",\"site\":\"\",\"count\":375,\"sum\":48214210142151},{\"filesystem\":
\"filesys1\",\"owner\":\"benjamin\",\"site\":\"\",\"count\":2,\"sum\":3140759161},
{\"filesystem\":\"filesys1\",\"owner\":\"babcock\",\"site\":\"\",\"count\":9082943,
\"sum\":48534270142857},{\"filesystem\":\"filesys1\",\"owner\":\"borgato\",
\"site\":\"\",\"count\":2,\"sum\":4415704787}]","count": 15,"doc_count": 376182496
}
```


In the following code block, the information from the rows field of the response is reflowed so that it is easier to read. The sum column is omitted. You can see that the rows are grouped by file system, owner, and site and are sorted by file system and owner:


```
"rows": "[
{\"filesystem\":\"fs10-1m-me1\", \"owner\":\"_NULL_\", \"site\":\"host\", \"count\":468,
{\"filesystem\":\"fs10-1m-me1\", \"owner\":\"nobody\", \"site\":\"host\", \"count\":4,
{\"filesystem\":\"fs10-1m-me1\", \"owner\":\"root\",   \"site\":\"host\", \"count\":1806,
{\"filesystem\":\"fs11-1m-me1\", \"owner\":\"_NULL_\", \"site\":\"host\", \"count\":521,
{\"filesystem\":\"fs11-1m-me1\", \"owner\":\"nobody\", \"site\":\"host\", \"count\":4,
{\"filesystem\":\"fs11-1m-me1\", \"owner\":\"root\",   \"site\":\"host\", \"count\":1854,
{\"filesystem\":\"filesys1\",    \"owner\":\"baldwin\",\"site\":\"\",     \"count\":2,
{\"filesystem\":\"filesys1\",    \"owner\":\"behr\",   \"site\":\"\",     \"count\":785144,
{\"filesystem\":\"filesys1\",    \"owner\":\"boston\", \"site\":\"\",     \"count\":4554,
{\"filesystem\":\"filesys1\",    \"owner\":\"beale\",  \"site\":\"\",     \"count\":203545,
{\"filesystem\":\"filesys1\",    \"owner\":\"root\",   \"site\":\"\",     \"count\":490,
{\"filesystem\":\"filesys1\",    \"owner\":\"broadwood\",\"site\":\"\",   \"count\":375,
{\"filesystem\":\"filesys1\",    \"owner\":\"benjamin\", \"site\":\",     \"count\":2,
{\"filesystem\":\"filesys1\",    \"owner\":\"babcock\",\"site\":\"\",     \"count\":9082943,
{\"filesystem\":\"filesys1\",    \"owner\":\"borgato\",\"site\":\"\",     \"count\":2,
```


2. The following example shows how to search for duplicate files with a size greater than 1 MiB:
   a. Step 1: Define the search parameters in a file named search.json:

```
{
    "query": "",
    "filters": [
      {
        "key": "size",
        "operator": ">",
        "value": 1048576
      }
    ],
    "group_by": ["size", "filename"],
    "sort_by": [{"size": "asc"}, {"filename": "asc"}],
    "limit": 100
}
```

   b. Step 2: Submit the request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ search -X POST -d@search.json
-H "Content-Type: application/json"
```

The following response is returned. Some of the rows of the response are omitted:


```
{
"facet_tree":{"OWNER": "[{\"owner\":\"behr\",\"count\":160096},{\"owner\":\"babcock\",
\"count\":14609},{\"owner\":\"beale\",\"count\":612},{\"owner\":\"root\",\"count\":34}]",
"DEPARTMENT":"[{\"department\":null,\"count\":175351}]","FILESYSTEM": "[{\"filesystem\":
\"filesys1\",\"count\":175351}]",    "PROJECT": "[{\"project\":\"TCGA_kirc\",\"count\":86},
{\"project\":\"TCGA_ucs\",\"count\":14},{\"project\":\"TCGA_stad\",\"count\":21},
{\"project\":\"TCGA_lusc\",\"count\":20},{\"project\":\"acc\",\"count\":2},{\"project\":
\"TCGA_meso\",\"count\":54},{\"project\":\"TCGA_skcm\",\"count\":107},{\"project\":
\"TCGA_ov\",\"count\":81},{\"project\":\"kich\",\"count\":10},{\"project\":\"TCGA_lgg\",
\"count\":137},{\"project\":\"TCGA_prad\",\"count\":662},{\"project\":\"TCGA_laml\",
\"count\":6},{\"project\":\"TCGA_thym\",\"count\":54},{\"project\":\"TCGA_thca\",\"count\":
255},{\"project\":\"kirp\",\"count\":256},{\"project\":\"hnsc\",\"count\":22},{\"project\":
\"TCGA_pcpg\",\"count\":2},{\"project\":\"Eichler\",\"count\":877},{\"project\":
\"TCGA_ucec\",\"count\":238},{\"project\":\"TCGA_luad\",\"count\":1417},{\"project\":
\"Level1\",\"count\":6},{\"project\":\"brca\",\"count\":14},{\"project\":null,\"count\":
168855},{\"project\":\"TCGA_paad\",\"count\":196},{\"project\":\"TCGA_read\",\"count\":244}
,{\"project\":\"cesc\",\"count\":613},{\"project\":\"coad\",\"count\":176},{\"project\":
\"dlbc\",\"count\":20},{\"project\":\"blca\",\"count\":471},{\"project\":\"TCGA_sarc\",
\"count\":90},{\"project\":\"TCGA_lihc\",\"count\":206},{\"project\":\"gbm\",\"count\":120},
{\"project\":\"esca\",\"count\":19}]","CLASSIFICATION": "[{\"classification\":null,
\"count\":175351}]","ARCHIVE": "[{\"archive\":null,\"count\":175351}]"},"query_time_secs":
1.377808,"rows": "[{\"size\":1048608,\"filename\":\"NA-C-ms22-cm0.mcdat\",\"count\":4106,
```

```
\"sum\":4305584448},{\"size\":1048608,\"filename\":\"asm_g100-C-ms22-cm0.mcdat\",
\"count\":2,\"sum\":2097216},{\"size\":1048608,\"filename\":\"asm_g1083-C-ms22-cm0.mcdat\",
\"count\":2,\"sum\":2097216},{\"size\":1048608,\"filename\":\"asm_g111-C-ms22-cm0.mcdat\",
\"count\":2,\"sum\":2097216},{\"size\":1048608,\"filename\":\"asm_g1122-C-ms22-cm0.mcdat\",
\"count\":2,\"sum\":2097216},{\"size\":1048608,\"filename\":\"asm_g134-C-ms22-cm0.mcdat\",
\"count\":2,\"sum\":2097216},{\"size\":1048608,\"filename\":\"asm_g145-C-ms22-cm0.mcdat\",
\"count\":2,\"sum\":2097216},{\"size\":1048608,\"filename\":\"asm_g147-C-ms22-cm0.mcdat\",
\"count\":3,\"sum\":3145824},{\"size\":1048608,\"filename\":\"asm_g149-C-ms22-cm0.mcdat\",
\"count\":4,\"sum\":4194432},{\"size\":1048608,\"filename\":\"asm_g151-C-ms22-cm0.mcdat\",
\"count\":2,\"sum\":2097216},{\"size\":1048608,\"filename\":\"asm_g153-C-ms22-cm0.mcdat\",
\"count\":2,
...
8583},{\"size\":1053910,\"filename\":\"asm_g2981.asm\",\"count\":2,\"sum\":2107820},
{\"size\":1053914,\"filename\":\"seqDB.v006.dat\",\"count\":2,\"sum\":2107828},
{\"size\":1054721,\"filename\":\"asm_g3384.asm\",\"count\":2,\"sum\":2109442}]","count":
100,"doc_count":4318
}
```


In the following code block, the information from the rows field of the response is reflowed so that it is easier to read. You can see that the rows are grouped by size and file name and that they are sorted by size and file name in ascending order:


```
"rows":
"[
{\"size\":1048608,\"filename\":\"NA-C-ms22-cm0.mcdat\",   \"count\":4106,\"sum\":4305584448},
{\"size\":1048608,\"filename\":\"asm_g100-C-ms22-cm0.mcdat\",   \"count\":2,\"sum\":2097216},
{\"size\":1048608,\"filename\":\"asm_g1083-C-ms22-cm0.mcdat\",  \"count\":2,\"sum\":2097216},
{\"size\":1048608,\"filename\":\"asm_g111-C-ms22-cm0.mcdat\",   \"count\":2,\"sum\":2097216},
{\"size\":1048608,\"filename\":\"asm_g1122-C-ms22-cm0.mcdat\",  \"count\":2,\"sum\":2097216},
{\"size\":1048608,\"filename\":\"asm_g134-C-ms22-cm0.mcdat\",   \"count\":2,\"sum\":2097216},
{\"size\":1048608,\"filename\":\"asm_g145-C-ms22-cm0.mcdat\",   \"count\":2,\"sum\":2097216},
{\"size\":1048608,\"filename\":\"asm_g147-C-ms22-cm0.mcdat\",   \"count\":3,\"sum\":3145824},
{\"size\":1048608,\"filename\":\"asm_g149-C-ms22-cm0.mcdat\",   \"count\":4,\"sum\":4194432},
{\"size\":1048608,\"filename\":\"asm_g151-C-ms22-cm0.mcdat\",   \"count\":2,\"sum\":2097216},
{\"size\":1048608,\"filename\":\"asm_g153-C-ms22-cm0.mcdat\",   \"count\":2,
...
{\"size\":1053910,\"filename\":\"asm_g2981.asm\",               \"count\":2,\"sum\":2107820},
{\"size\":1053914,\"filename\":\"seqDB.v006.dat\",              \"count\":2,\"sum\":2107828},
{\"size\":1054721,\"filename\":\"asm_g3384.asm\",               \"count\":2,\"sum\":2109442}
]",
```


3. The following example shows how to perform a nongrouped search (record level results) for files owned by **benjamin**:
   a. Step 1: Define the search parameters in a file named search.json:

```
{
    "query": "",
    "filters": [
      {
        "key": "owner",
        "operator": "=",
        "value": "benjamin"
      }
    ],
    "group_by": [],
    "sort_by": [],
    "limit": 100
}
```

   b. Step 2: Submit the following request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ search -X POST -d@search.json
-H "Content-Type: application/json"
```

The following response is returned:


```
* Connection #0 to host localhost left intact
{
"query_time_secs": 1422.651552,"rows": "[{\"filesystem\":\"filesys1\",\"revision\":\"MO1\",
\"site\":\"\",\"platform\":\"Spectrum Scale\",\"cluster\":\"filesys1.university.edu\",
\"inode\":257173,\"owner\":\"benjamin\",\"group\":\"iacs\",\"permissions\":\"-r--r--r--\",
\"fileset\":\"hg19\",\"uid\":null,\"gid\":null,\"path\":\"\\/filesys1\\/hg19\\/\",
\"filename\":\"Homo_sapiens_assembly19.fasta.fai\",\"filetype\":\"fasta\",\"migstatus\":
\"resdnt\",\"migloc\":\"NA\",\"mtime\":\"2014-08-08T19:37:13.000Z\",\"atime\":
\"2017-08-27T22:21:11.000Z\",\"ctime\":\"2016-02-22T19:54:30.000Z\",\"inserttime\":
\"2018-08-02T15:47:25.000Z\",\"tier\":\"system\",\"size\":2780,\"qpart\":1,\"fkey\":
\"filesys1.university.edufilesys1257173\",\"project\":null,\"department\":null,
\"archive\":null,\"classification\":null,\"tag5\":null,\"tag6\":null,\"tag7\":null,
\"tag8\":null,\"tag9\":null,\"tag10\":null,\"tag11\":null,\"tag12\":null,\"tag13\":null,
\"tag14\":null,\"tag15\":null,\"tag16\":null},{\"filesystem\":\"filesys1\",\"revision\":
\"MO1\",\"site\":\"\",\"platform\":\"Spectrum Scale\",\"cluster\":
\"filesys1.university.edu\",\"inode\":322176,\"owner\":\"benjamin\",\"group\":\"iacs\",
\"permissions\":\"-r--r--r--\",\"fileset\":\"hg19\",\"uid\":null,\"gid\":null,\"path\":
\"\\/filesys1\\/hg19\\/\",\"filename\":\"Homo_sapiens_assembly19.fasta\",\"filetype\":
\"fasta\",\"migstatus\":\"resdnt\",\"migloc\":\"NA\",\"mtime\":
\"2014-08-08T19:37:13.000Z\",\"atime\":\"2017-08-27T22:21:15.000Z\",\"ctime\":
\"2016-02-22T19:44:09.000Z\",\"inserttime\":\"2018-08-02T15:47:25.000Z\",\"tier\":
\"system\",\"size\":3140756381,\"qpart\":4,
\"fkey\":\"filesys1.university.edufilesys1322176\",\"project\":null,\"department\":null,
\"archive\":null,\"classification\":null,\"tag5\":null,\"tag6\":null,\"tag7\":null,\"tag8\":null,
\"tag9\":null,\"tag10\":null,\"tag11\":null,\"tag12\":null,\"tag13\":null,\"tag14\":null,
\"tag15\":null,\"tag16\":null}]","doc_count": 2,"count": 2,"facet_tree": {"OWNER":
 "[{\"owner\":\"benjamin\",\"count\":2.0}]","FILESYSTEM": "[{\"filesystem\":\"filesys1\",
\"count\":2.0}]","ARCHIVE": "[{\"archive\":null,\"count\":2.0}]","CLUSTER":
"[{\"cluster\":\"filesys1.university.edu\",\"count\":2.0}]","SITE": "[{\"site\":\"\",
\"count\":2.0}]","CLASSIFICATION": "[{\"classification\":null,\"count\":2.0}]",
```

```
"DEPARTMENT": "[{\"department\":null,\"count\":2.0}]","PLATFORM": "[{\"platform\":
\"Spectrum Scale\",\"count\":2.0}]","TIER": "[{\"tier\":\"system\",\"count\":2.0}]",
"PROJECT": "[{\"project\":null,\"count\":2.0}]"}
}
```

In the following code block, the information from the rows field of the response is reflowed for better viewing. Many columns are omitted. You can see that the response returns two rows in which the owner field is benjamin:

```
"rows": "[
{\"filesystem\":\"filesys1\",\"revision\":\"MO1\",\"site\":\"\", ...\"owner\":\"benjamin\", ...
{\"filesystem\":\"filesys1\",\"revision\":\"MO1\",\"site\":\"\", ...\"owner\":\"benjamin\", ...}
]",
```

/db2whrest/v1/report: POST

Creates a curation report definition and runs it immediately.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/report -X POST -d@report.json -H
"Content-Type: application/json"
```

This endpoint has the same parameters as the `/db2whrest/v1/search` endpoint and also has a name parameter:

query
: Specifies a search query string.

filters
: Specifies an array of dictionaries that filter the results of the query. Each dictionary must contain the following three fields:

  key
  : The name of the field or column to be returned.

  operator
  : One of the following operators: `=, >, <, <>, <=, >=, is, like`.

  value
  : The value of the field or column to filter on.

group_by
: Specifies a list of fields to be used to summarize search results. Grouped queries return output columns for count and sum, whereas nongrouped queries return all columns for each row. If the `group_by` field is not specified, the search returns record-level information.

sort_by
: Specifies an array of dictionary objects. Each dictionary object must specify a field or column name to be sorted on and a sort direction. Valid sort directions are asc and desc.

limit
: Specifies the maximum number of rows that can be returned by the response.

name
: Specifies a name for the report output file. If this parameter is not specified, the endpoint uses the randomly generated UUID as the file name base.

The following example illustrates how to specify these fields:

```
{
    "name": "Unassigned Project Report",
    "query": "platform='Spectrum Scale'",
    "filters": [
      {
        "key": "project",
        "operator": "is",
        "value": "null"
      }
    ],
    "group_by": ["Filesystem","Owner","Site"],
    "sort_by": [{"Filesystem": "asc"},{"Owner": "asc"}],
    "limit": 100000
}
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Status codes

201
: The operation is successful.

All other status code values
: The operation failed.

- 201: The operation is successful.
- All other status code values: The operation failed.

Examples

1. The following example shows how to create a report:
   a. Step 1: Define the search parameters in a file named report.json:

```
{
    "name": "Unassigned Project Report",
    "query": "platform='Spectrum Scale'",
    "filters": [
      {
        "key": "project",
        "operator": "is",
        "value": "null"
      }
    ],
    "group_by": ["Filesystem","Owner","Site"],
    "sort_by": [{"Filesystem": "asc"},{"Owner": "asc"}],
    "limit": 100000
}
```

   b. Step 2: Submit the following request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/report -X POST -d@report.json
-H "Content-Type: application/json"
```

2. The response includes the ID of the new report and the status of the operation ("report created"):

```
{"report": "b5ff3126-353d-4d7d-857a-750cc20b8bab", "status": "report created"}
```

Report scripts

IBM Data Cataloging contains a set of predefined JSON and scripts to generate specific reports.

1. JSON Examples
   - The following example lists the files or objects that are accessed in last 0 - 30 days. This report is created in the UI.

     age_report_0-30_days_since_access_detail.json

```
{
    "name": "Age Report Detail. 0-30 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 30 DAYS"
      }
    ],
    "group_by": [],
    "sort_by": []
}
```

   - This example summarizes the files or objects that are accessed in last 0 - 30 days and are grouped by data source. This report is created in the UI.

     age_report_0-30_days_since_access_summary.json

```
{
    "name": "Age Report Summary. 0-30 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 30 DAYS"
      }
    ],
    "group_by": ["datasource"],
    "sort_by": [{"datasource": "asc"}]
}
```

   - This example lists the files or objects that are accessed in last 30 - 60 days. This report is created in the UI.

     age_report_30-60_days_since_access_detail.json

```
{
    "name": "Age Report Detail. 30-60 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 30 DAYS"
      },
```

```
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 60 DAYS"
      }
    ],
    "group_by": [],
    "sort_by": []
}
```

- This example summarizes the files or objects that are accessed in last 30 - 60 days and are grouped by data source. This report is created in the UI.

  age_report_30-60_days_since_access_summary.json

```
{
    "name": "Age Report Summary. 30-60 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 30 DAYS"
      },
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 60 DAYS"
      }
    ],
    "group_by": ["datasource"],
    "sort_by": [{"datasource": "asc"}]
}
```

- This example lists the files or objects that are accessed in last 60 - 90 days. This report is created in the UI.

  age_report_60-90_days_since_access_detail.json

```
{
    "name": "Age Report Detail. 60-90 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 60 DAYS"
      },
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 90 DAYS"
      }
    ],
    "group_by": [],
    "sort_by": []
}
```

- The following example summarizes the files or objects that are accessed in last 60 - 90 days and are grouped by data source. This report is created in the UI.

  age_report_60-90_days_since_access_summary.json

```
{
    "name": "Age Report Summary. 60-90 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 60 DAYS"
      },
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 90 DAYS"
      }
    ],
    "group_by": ["datasource"],
    "sort_by": [{"datasource": "asc"}]
}
```

- The following example lists the files or objects that are accessed in last 90 - 180 days. This report is created in the UI.

  age_report_90-180_days_since_access_detail.json

```
{
    "name": "Age Report Detail. 90-180 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 90 DAYS"
      },
      {
        "key": "atime",
        "operator": ">",
```

```
        "value": "NOW() - 180 DAYS"
      }
    ],
    "group_by": [],
    "sort_by": []
}
```

- The following example summarizes the files or objects that are accessed in last 90 - 180 days and are grouped by data source. This report is created in the UI.

  age_report_90-180_days_since_access_summary.json

```
{
    "name": "Age Report Summary. 90-180 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 90 DAYS"
      },
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 180 DAYS"
      }
    ],
    "group_by": ["datasource"],
    "sort_by": [{"datasource": "asc"}]
}
```

- The following example lists the files or objects that are accessed in last 180 - 360 days. This report is created in the UI.

  age_report_180-360_days_since_access_detail.json

```
{
    "name": "Age Report Detail. 180-360 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 180 DAYS"
      },
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 360 DAYS"
      }
    ],
    "group_by": [],
    "sort_by": []
}
```

- The following example summarizes the files or objects that are accessed in last 180 - 360 days and are grouped by data source. This report is created in the UI.

  age_report_180-360_days_since_access_summary.json

```
{
    "name": "Age Report Summary. 180-360 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 180 DAYS"
      },
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 360 DAYS"
      }
    ],
    "group_by": ["datasource"],
    "sort_by": [{"datasource": "asc"}]
}
```

- The following example lists the files or objects that are accessed in last 360 - 720 days. This report is created in the UI.

  age_report_360-720_days_since_access_detail.json

```
{
    "name": "Age Report Detail. 360-720 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 360 DAYS"
      },
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 720 DAYS"
```

```
      }
    ],
    "group_by": [],
    "sort_by": []
}
```

- The following example summarizes the files or objects that are accessed in last 360 - 720 days and are grouped by data source. This report is created in the UI.

  age_report_360-720_days_since_access_summary.json

```
{
    "name": "Age Report Summary. 360-720 Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 360 DAYS"
      },
      {
        "key": "atime",
        "operator": ">",
        "value": "NOW() - 720 DAYS"
      }
    ],
    "group_by": ["datasource"],
    "sort_by": [{"datasource": "asc"}]
}
```

- The following example lists the files or objects accessed that are not accessed in the last 720 days. This report is created in the UI.

  age_report_720+_days_since_access_detail.json

```
{
    "name": "Age Report Detail. 720+ Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 720 DAYS"
      }
    ],
    "group_by": [],
    "sort_by": []
}
```

- The following example summarizes the files or objects that are accessed in last 720 days and are grouped by data source. This report is created in the UI.

  age_report_720+_days_since_access_summary.json

```
{
    "name": "Age Report Summary. 720+ Days",
    "query": "",
    "filters": [
      {
        "key": "atime",
        "operator": "<=",
        "value": "NOW() - 720 DAYS"
      }
    ],
    "group_by": ["datasource"],
    "sort_by": [{"datasource": "asc"}]
}
```

2. SQL scripts
   - This script provides the count of potentially duplicate files across the heterogeneous environment. This report is not created in the UI.

     duplicate_files_by_count.sql

```
select filename, size, count(fkey) from metaocean group by filename,
    size order by count(fkey) desc limit 20;
```

   - This script provides the size of potentially duplicate files across the heterogeneous environment. This report is not created in the UI.

     duplicate_files_by_total_size.sql

```
select filename,entrysize,entrycount,totalsize from (select filename,
    size as entrysize, count(fkey) as entrycount, count(fkey)*size as TotalSize
    from metaocean group by filename,size) where entrycount>1 order by totalsize desc;
```

   - This script provides the size of potentially duplicate files across the heterogeneous environment. This report is not created in the UI.

     size_snap.sql

```
select datasource,count(*),sum(size)/1024/1024/1024,max(mtime),max(atime)
    from metaocean group by datasource with ur
```

   - This script provides a view of the capacity that is consumed per collection. This report is not created in the UI.

     space_per_collection.sql

```
select metaocean.collection,count(*),sum(size)/1024/1024/1024,max(mtime),datasource,
    tier from metaocean group by metaocean.collection,datasource,tier order by max(mtime)
```

```
    desc with ur
```

- This script provides a view of the capacity that is consumed per file type. This report is not created in the UI.

  space_per_filetype.sql

  ```
  select filetype,datasource,tier,count(*),sum(size)/1024/1024/1024 from metaocean
      group by filetype,datasource,tier order by filetype,datasource,tier desc with ur
  ```

- This script provides a view of the capacity that is consumed per user. This report is not created in the UI.

  space_per_user.sql

  ```
  select owner,tier,count(*),sum(size)/1024/1024/1024,max(mtime),max(atime) from metaocean
      group by owner,tier order by owner,tier with ur
  ```

Reports can also be generated by using one of two CLI utilities that are provided with IBM Data Cataloging. Both exist on the IBM Data Cataloging nodes:

- /opt/ibm/metaocean/reports/generate_report.py
  To run the utilities, log in to an IBM Data Cataloging node and run generate_report.py with the following usage:

  ```
  python generate_report.py [-h] [-o filename] -u username [infile]
  ```

  - Run a report with an SQL input file:

    ```
    python generate_report.py -u sdadmin -o report.csv sql/space_per_user.sql
    Enter password for SD user 'sdadmin':
    ```

  - Run a report with a JSON input file:

    ```
    python generate_report.py -u sdadmin  sql/age_report_0-30_days_since_access_detail.json
    Enter password for SD user 'sdadmin':
    ```

  This requires a IBM Data Cataloging user name and input file. You must have the Data Admin role to generate reports. The tool prompts for a password

  Input file examples are stored in the /opt/ibm/metaocean/reports/sql directory. There is a mix of JSON and SQL files in the directory. All JSON files create a report in the IBM Data Cataloging UI. The SQL files create a CSV file.

- /opt/ibm/metaocean/reports/generate_path_age_report.py
  Important: This tool does not create reports in the IBM Data Cataloging UI.
  Here is the usage of the tool:

  ```
  generate_path_age_report.py [-h] -u username -r report -p pathlevel -a archive

  optional arguments:
    -h, --help            show this help message and exit
    -u username, --user username
                          User name with authority to create reports
    -r report, --report report
                          The report type to be generated (ARDS, ARDSOW, AROW, ARPL,
                          CPPL, FTMB, CPFT, FTMB50)
    -p pathlevel, --pathlevel pathlevel
                          The level of path used in some reports
    -a archive, --archive archive
                          The archive threshold in months
  ```

  The `generate_path_age_report.py` tool needs a minimum of three parameters:
  - `pathlevel`
    The level of path used in some reports, for example:

    ```
    1 = /x/, 2=/x/y/
    ```

    If your report does not use this parameter a value of 1 must be used.
  - `archive`
    The archive threshold is the number of months since the last time a file is considered relevant for archiving for reporting purposes. If your report does not use this parameter a value of 1 must be used.

  - `report`
    The report type is one of the following codes:

    ```
    ARDS   = Summary of archivable capacity grouped by datasource
    ARDSOW = Summary of archivable capacity grouped by datasource
    AROW   = Summary of archivable capacity grouped by owner
    ARPL   = Summary of archivable capacity grouped by specified path level
    CPPL   = Summary of capacity grouped by specified path level
    FTMB   = Summary of filetype usage by month, previous 12 months
    CPFT   = Summary of capacity grouped by filetype
    FTMB50 = Summary of filetype usage by month, previous 12 months' top 50 filetypes
    ```

Here is an example of running the `generate_path_age_report.py` tool:

```
python generate_path_age_report.py -u sdadmin -r ARDSOW -p 2 -a 12
Starting to create report type 'ARDSOW' for user: sdadmin
Enter password for SD user 'sdadmin':
Setting up path level summary table
Generating path level summary table
Generating path level summary table complete.
Generating report
Generating report - building temporary table.
Generating report - querying temporary table.
Generating report - writing output file.
Report generated successfully. Results are in 'rpt_ar_ds_ow.csv'.
```

/db2whrest/v1/report: GET

Gets information on all the curation reports that are available in the system.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ report -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

1. The following example shows how to get information on all the curation reports that are available in the system.
   a. Submit the request:

   ```
   curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/ v1/report -X GET -H "Accept: application/json"
   ```

   b. The following response is returned. In this example the system contained a total of only two reports, and the response displays the available information about both reports. The response data is in JSON format. The data has been manually reflowed to make it more readable.

   ```
   [
     {
       "name": "Age Report Detail. 0-30 Days",
       "query": "{\"filters\": [{\"value\": \"NOW() - 30 DAYS\", \"key\": \"atime\", \"operator\":
       \">\"}], \"sort_by\": [], \"name\": \"Age Report Detail. 0-30 Days\", \"query\": \"\",
       \"group_by\": []}",
       "size": 208,
       "report": "34d03123-4611-4f9f-9730-ffe89c6c53cb",
       "duration": 0,
       "filename": "Age Report Detail. 0-30 Days-2018-11-01_18:58:04.csv",
       "lastrun": "2018-11-01T18:58:04.000Z",
       "status": "complete",
       "schedule": null
     },
     {
       "name": "Age Report Summary. 0-30 Days",
       "query": "{\"filters\": [{\"value\": \"NOW() - 30 DAYS\", \"key\": \"atime\", \"operator\":
       \">\"}], \"sort_by\": [{\"datasource\": \"asc\"}], \"name\": \"Age Report Summary. 0-30 Days\",
       \"query\": \"\", \"group_by\": [\"datasource\"]}",
       "size": 21,
       "report": "b5ff3126-353d-4d7d-857a-750cc20b8bab",
       "duration": 0,
       "filename": "Age Report Summary. 0-30 Days-2018-11-01_18:58:14.csv",
       "lastrun": "2018-11-01T18:58:14.000Z",
       "status": "complete",
       "schedule": null
     }
   ]
   ```

Note: To download the output of a report, use the download endpoint. For more information, see /db2whrest/v1/report/<report_id>/download: GET.

/db2whrest/v1/report/<report_id>: GET

Gets information about a single curation report.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/report/ <report_id> -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

1. The following example shows how to get information about a single curation report.
   a. Submit a request to display information about report ID "b5ff3126-353d-4d7d-857a-750cc20b8bab":

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ report/b5ff3126-353d-4d7d-
857a-750cc20b8bab -X GET -H "Accept: application/json"
```

   b. The following response is returned. The response data is in JSON format:

```
{"name": "Age Report Summary. 0-30 Days", "query":"{\"filters\":[{\"value\":\"NOW() -
30 DAYS\",\"key\":\"atime\",\"operator\":\">\"}],\"sort_by\":[{\"datasource\":\"asc\"}],
\"name\":\"Age Report Summary. 0-30 Days\",\"query\":\"\",\"group_by\":[\"datasource\"]}",
"size":21,"report":"b5ff3126-353d-4d7d-857a-750cc20b8bab","duration":0,"filename":"Age
Report Summary. 0-30 Days-2018-11-01_18:58:14.csv","lastrun":"2018-11-01T18:58:14.000Z",
"status":"complete","schedule":null}
```

   In the following code block, the response data has been manually reflowed to make it easier to read:

```
{
"name":"Age Report Summary. 0-30 Days","query":
"{\"filters\":[{\"value\":\"NOW() - 30 DAYS\",\"key\":\"atime\",\"operator\":\">\"}],
\"sort_by\":[{\"datasource\":\"asc\"}],\"name\":\"Age Report Summary. 0-30 Days\",
\"query\":\"\",\"group_by\":[\"datasource\"]}",
"size":21,
"report":"b5ff3126-353d-4d7d-857a-750cc20b8bab",
"duration":0,
"filename":"Age Report Summary. 0-30 Days-2018-11-01_18:58:14.csv",
"lastrun":"2018-11-01T18:58:14.000Z",
"status":"complete",
"schedule":null
}
```

   Note: To download the output of a report, use the download endpoint. For more information, see /db2whrest/v1/report/<report_id>/download: GET.

/db2whrest/v1/report/<report_id>/download: GET

Downloads the output file of the specified curation report.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/report/ <report_id>/download -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- CSV

Status codes

- 200: The operation is successful.
- Any other status code value: The operation failed.

Examples

1. The following example shows how to download the output file of a specific report.
   a. Submit a request to display the output from report ID "6baae44c-2b85-4954-ba44-d3e637d4b48d":

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ report/6baae44c-2b85-4954-
ba44-d3e637d4b48d/download
```

This report searches for files greater than 1 MiB in size and grouped by datasource.

b. The following response is returned. The output is in CSV format:

```
datasource,count,sum
filesystem1,346.0,685733316581
filesystem2,370213.0,249855575516039
```

/db2whrest/v1/bucket: GET

Gets information on all buckets that are supported.

Buckets influence the manner in which IBM Data Cataloging groups data for search faceting and in the user interface widgets. The predefined bucket types that are supported are:

- SizeRange
- TimeSinceAccess
- FileGroup

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/buckets -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

1. The following example shows how to get information on all the buckets that are supported by the bucket service.
   a. Submit the request:

```
curl -k -H 'Authorization: Bearer <token> https://<data_cataloging_host>/db2whrest/v1/buckets -X GET -H "Accept:
application/json"
```

   b. The following response is returned.

```
[
  {
    "data": "{\"type\": \"int\", \"source_field\": \"size\", \"ranges\": {\"extra small\": [0, 4096], \"small\":
[4096, 1048576], \"extra large\": [1099511627776, \"INFINITY\"], \"large\": [1073741824, 1099511627776], \"medium\":
[1048576, 1073741824]}, \"name\": \"SizeRange\"}",
    "name": "SizeRange"
  },
  {
    "data": "{\"ranges\": {\"1 year+\": [365, \"INFINITY\"], \"1 quarter\": [30, 90], \"1 year\": [90, 365], \"1
month\": [7, 30], \"1 week\": [0, 7]}, \"source_field\": \"atime\", \"type\": \"date\", \"name\":
\"TimeSinceAccess\"}",
    "name": "TimeSinceAccess"
  },
  {
    "name": "FileGroup",
    "data": "{\"name\": \"FileGroup\", \"type\": \"list\", \"ranges\": {\"pdf\": [\"pdf\"], \"doc\": [\"doc\",
\"docx\", \"odt\"], \"xls\": [\"xls\", \"xlsm\", \"xlsx\", \"ods\"], \"image\": [\"jpg\", \"jpeg\", \"png\", \"gif\",
\"tif\"], \"ppt\": [\"pptx\", \"ppt\", \"pps\", \"odp\"], \"compressed\": [\"zip\", \"tar\", \"gz\", \"z\", \"7z\",
\"xz\"], \"text\": [\"txt\", \"rtf\"], \"audio\": [\"mp3\", \"wav\"], \"video\": [\"mp4\", \"mov\", \"mpg\",
\"mkv\"], \"medical_image\": [\"img\", \"hdr\", \"nii\", \"mnc\", \"dcm\"], \"genomics\": [\"bam\", \"sam\",
\"vcf\"], \"semi_structured\": [\"csv\", \"tsv\", \"avro\", \"json\", \"xml\", \"parquet\"]}, \"source_field\":
\"filetype\"}"
  }
]
```

/db2whrest/v1/buckets/<bucket>: GET

Gets information on a specific bucket that is supported.

Buckets influence the manner in which IBM Data Cataloging groups data for search faceting and in the user interface widgets. The predefined bucket types that are supported are:

- SizeRange
- TimeSinceAccess
- FileGroup

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/buckets/<bucket> -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

1. The following example shows how to get information on all the buckets that are supported by the bucket service.
   a. Submit the request:

   ```
   curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/buckets/SizeRange -X GET -H
   "Accept: application/json"
   ```

   b. The following response is returned.

   ```
   [
     {
       "data": "{\"type\": \"int\", \"source_field\": \"size\", \"ranges\": {\"extra small\": [0, 4096], \"small\":
   [4096, 1048576], \"extra large\": [1099511627776, \"INFINITY\"], \"large\": [1073741824, 1099511627776], \"medium\":
   [1048576, 1073741824]}, \"name\": \"SizeRange\"}",
       "name": "SizeRange"
     }
   ]
   ```

/db2whrest/v1/report/<report_id>: PUT

Runs the specified curation report. The old output file is replaced by the new one.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/report/ <report_id> -X PUT
```

Supported request types and response formats

Supported request types:

- PUT

Supported response formats:

- JSON

/db2whrest/v1/buckets/<bucket>: PUT

Modifies the bucket details.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	X	X

To modify the details of a bucket, you must create a JSON object that is modeled after the original (or existing) bucket definition as shown in the sample:

```
{
  "name": "SizeRange",
  "source_field": "size",
  "type": "int"
  "ranges": {
    "extra small": [ 0, 4096 ],
    "small": [ 4096, 1048576 ],
    "medium": [ 1048576, 1073741824 ],
    "large": [ 1073741824, 1099511627776 ],
    "extra large": [ 1099511627776, "INFINITY" ]
  }
}
```

```
{
  "name": "TimeSinceAccess",
  "source_field": "atime",
  "type": "date",
  "ranges": {
    "1 week\": [ 0, 7 ],
    "1 month": [ 7, 30 ],
    "1 quarter": [ 30, 90 ],
    "1 year": [ 90, 365 ],
    "1 year+": [ 365, "INFINITY" ]
    }
}
```

```
{
  "name": "FileGroup",
  "type": "list",
  "ranges": {
    "pdf": [ "pdf" ],
    "doc": [ "doc", "docx", "odt" ],
    "xls": [ "xls", "xlsm", "xlsx", "ods" ],
    "image": [ "jpg", "jpeg", "png", "gif", "tif" ],
    "ppt": [ "pptx", "ppt", "pps", "odp" ],
    "compressed": [ "zip", "tar", "gz", "z", "7z", "xz" ],
    "text": [ "txt", "rtf" ],
    "audio": [ "mp3", "wav" ],
    "video": [ "mp4", "mov", "mpg", "mkv" ],
    "medical_image": [ "img", "hdr", "nii", "mnc", "dcm" ],
    "genomics": [ "bam", "sam", "vcf" ],
    "semi_structured": [ "csv", "tsv", "avro", "json", "xml", "parquet" ]
  },
  "source_field": "filetype"
}
```

The JSON object must be based on the default values of the bucket. You can modify the range labels or values. The bucket name, source_field, and bucket types cannot be modified.

The three bucket types that are supported include:

int
: For the "int" types, each range consists of a label and a minimum and maximum value. The only 'int' type bucket that is supported currently is *SizeRange*. The units for 'SizeRange' are in bytes. A sample int type appears as shown:

  ```
  "small": [4096, 8192]
  ```

date
: For the "date" types, each range consists of a label and a minimum and maximum value that represents the number of days. The only 'date' type bucket that is supported in the current version is *TimeSinceAccess*. A sample date range appears as shown:

  ```
  "1_to_2_years": [365, 730]
  ```

list
: For the "list" types, each range consists of a label and a list of at least one string value. The only 'list' type bucket that is supported in the current version is *FileGroup*. The list of values must be a list of file extensions to be grouped. A sample list range appears as shown:

  ```
  "images": ["jpg", "jpeg", "gif", "png", "bmp"]
  ```

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/buckets/<bucket> -X PUT -d@bucket.json -
H "Content-Type: application/json
```

Supported request types and response formats

Supported request types:

- PUT

Supported response formats:

- JSON

Examples

The following example shows how to modify information for a single bucket.

1. Submit the request:

```
curl -k -H 'Authorization: Bearer <token>' https://<master_node>/db2whrest/v1/buckets/SizeRange -X PUT -d@bucket.json -H
"Content-Type: application/json
```

2. The following response is returned:

```
[
  {
    "status": "success"
  }
]
```

/db2whrest/v1/report/<report_id>: DELETE

Deletes the specified curation report definition and its output file.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/report/ <report_id> -X DELETE
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

/db2whrest/v1/sql_query?<sql>: GET

Retrieves data from a database with an SQL query in a GET command. You can use any standard SQL query. You must include the SQL query as a parameter of the URL. To send the query in an HTTP message body, see /db2whrest/v1/sql_query: POST.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	√[1]	√[1]	X	X
[1]The search is restricted to documents that are tagged with collections to which the user has a datauser role assigned.				

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/sql_query?<sql_statements> -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- CSV
- JSON

Examples

1. The following example shows how to get data from a database with an SQL query in a GET command. The SQL query is included as a parameter of the URL. This example specifies that the data must be returned in JSON format:
   a. Submit the request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/sql_query?
select%20*%20from%20meteor%20limit%202 -X GET -H "Accept: application/json"
```

   b. The following response is returned:

```
[{"datasource":"ctolib","operation":"INDEX","ingesttype":"SCAN","revision":"MO1","site":
"","platform":"Spectrum Scale","cluster":"ctolib.cluster1","inode":4042,"owner":"user1",
"group":"group1","permissions":"-rw-r--r--","fileset":"brother","uid":null,"gid":null,
"path":"\/ctolib\/brother\/dir1\/","filename":"file1.txt","filetype":"txt",
"recordversion":"","migstatus":"resdnt","migloc":"NA","mtime":"2015-08-20T21:47:09.000Z",
"atime":"2016-06-08T17:20:41.480Z","ctime":"2016-03-12T15:12:49.565Z","inserttime":
"2018-10-24T04:19:37.000Z","updatedtime":"2018-10-24T04:19:37.000Z","requesttime":
"2018-10-24T04:19:37.000Z","scangen":1,"tier":"system","size":28185,"qpart":0,"fkey":
"ctolib.cluster1ctolib4042","collection":null,"temperature":null,"duplicate":null,
"some-tag":null,"tag2":null,"tag3":null,"tag4":null,"tag5":null,"tag6":null,"tag7":null,
"tag8":null,"tag9":null,"tag10":null,"tag11":null,"tag12":null,"tag13":null,"tag14":null,
"tag15":null,"tag16":null},
{"datasource":"ctolib","operation":"INDEX","ingesttype":"SCAN","revision":"MO1","site":
"","platform":"Spectrum Scale","cluster":"ctolib.cluster1","inode":285102,"owner":"user1",
"group": "group1","permissions":"-rw-r--r--","fileset":"brother","uid":null,"gid":null,
"path":"\/ctolib\/brother\/dir1\/","filename":"file2.txt","filetype":"txt",
"recordversion":"","migstatus":"resdnt","migloc":"NA","mtime":"2015-08-21T00:55:33.000Z",
"atime":"2016-06-24T00:51:00.797Z","ctime":"2016-03-10T22:55:06.160Z","inserttime":
"2018-10-24T04:19:50.000Z","updatedtime":"2018-10-24T04:19:50.000Z","requesttime":
"2018-10-24T04:19:50.000Z","scangen":1,"tier":"system","size":435947,"qpart":0,"fkey":
"ctolib.cluster1ctolib28510 2","collection":null,"temperature":null,"duplicate":null,
"some-tag":null,"tag2":null,"tag3":null,"tag4":null,"tag5":null,"tag6":null,"tag7":null,
"tag8":null,"tag9":null,"tag10":null,"tag11":null,"tag12":null,"tag13":null,"tag14":
null,"tag15":null,"tag16":null}]
```


2. The following example makes the same query as the preceding example but does not specify an output format:
   a. Submit the request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/sql_query?
select%20*%20from%20meteor%20limit%202 -X GET"
```

   b. The following response is returned. The data is in CSV format:


```
0,ctolib,INDEX,SCAN,MO1,,Spectrum Scale,ctolib.cluster1,250692,user1,group1,-rw-r--r--,
brother,,,/ctolib/brother/dir1/,file1.txt,txt,,resdnt,NA,2015-08-20 11:26:29.000000,
2016-06-2323:26:00.113925,2016-03-10 19:33:36.639688,2018-10-24 04:19:50.000000,
2018-10-2404:19:50.000000,2018-10-24 04:19:50.000000,1,system,266314,0,
ctolib.cluster1ctolib250692,
,,,,,,,,,,,,,,,,,,
1,ctolib,INDEX,SCAN,MO1,,Spectrum Scale,ctolib.cluster1,250662,user1,group1,-rw-r--r--,
brother,,,/ctolib/brother/dir1,file2..txt,txt,,resdnt,NA,2015-08-20 11:26:25.000000
,2016-06-23 23:25:59.990565,2016-03-10 19:33:36.070314,2018-10-24 04:19:50.000000,
2018-10-24 04:19:50.000000,2018-10-24 04:19:50.000000,1,system,305365,0,
ctolib.cluster1ctolib250662,
,,,,,,,,,,,,,,,,,,
```


/db2whrest/v1/sql_query: POST

Gets data from a database with an SQL query in a POST command. You can use any standard SQL query. You must include the SQL query in the message body of the POST command.
To send the query as a parameter of the URL, see /db2whrest/v1/sql_query?<sql>: GET. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	√[1]	√[1]	X	X
[1]The search is restricted to documents that are tagged with collections to which the user ID has a datauser role assigned.				

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token> https://<data_cataloging_host>/db2whrest/v1/ sql_query -X POST -d@<sql.dat>
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- CSV
- JSON

Examples

1. The following example shows how to get the data from a database with an SQL query in a POST command. The SQL query is sent in the POST message body. The output format is plain text.
   a. Write the record to be added to the database into a file named sql.dat.
   b. Submit the request:

```
curl -k -H 'Authorization: Bearer <token> https://<data_cataloging_host> /db2whrest/v1/sql_query -X POST -d"select *
from datasrv04 where owner='root'"
```

   c. The following response is returned. The output is in plain text format:

```
0,filesys1,INDEX,SCAN,MO1,,Spectrum Scale,filesys1.university.edu,3067343,root,root,
-rw-r--r--,root,9,10,/filesys1/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,
sjdbList.out.tab,tab,resdnt,NA,2016-11-14 19:40:10,2017-06-02 21:30:26,
2017-06-02 21:30:26,2018-07-24 16:32:47,system,0,1,filesys1.university.edufilesys13067343,
,,,,,,,,,,,,,,,
1,filesys1,INDEX,SCAN,MO1,,Spectrum Scale,filesys1.university.edu,3067333,root,root,
-rw-r--r--,root,9,10,/filesys1/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,
chrLength.txt,txt,resdnt,NA,2016-11-14 19:40:10,2017-06-02 23:28:21,2017-06-02 21:30:26,
2018-07-24 16:32:47,system,412,1,filesys1.university.edufilesys13067333,
,,,,,,,,,,,,,,,,
...
```

In the following code block, the information from the response is curtailed so that it is easier to read. Only the first 14 columns are shown here. You can see that both rows seem to have root as the owner:

```
0,filesys1,INDEX,SCAN,MO1,,Spectrum Scale,filesys1.university.edu,3067343,root,root,
-rw-r--r--,root,9,10,/filesys1/. . . .
1,filesys1,INDEX,SCAN,MO1,,Spectrum Scale,filesys1.university.edu,3067333,root,root,
-rw-r--r--,root,9,10,/filesys1/. . . .
...
```

2. The following example shows how to get the data from a database with an SQL query in a POST command. The SQL query is sent in the POST message body. The output format is JSON.
   a. Write the SQL record in a file named sql.dat.
   b. Submit the request.

```
curl -k -H 'Authorization: Bearer <token> https://<data_cataloging_host>/db2whrest/v1/ sql_query -X POST -d@"sql.dat"
-H "accept: application/json"
```

   c. The following response is returned. The output is in JSON format:


```
[{"filesystem":"filesys1","operation":"INDEX","source":"SCAN","revision":"MO1","site":"",
"platform":"Spectrum Scale","cluster":"filesys1.university.edu","inode":3067112,"owner":
"root","group":"root","permissions":"-r--r--r--","fileset":"root","uid":9,"gid":10,"path":
"\/filesys1\/blast_db\/","filename":"other_genomic.75.tar.gz","filetype":"gz","migstatus":
"resdnt","migloc":"NA","mtime":"2014-12-31T06:00:00.000Z","atime":
"2016-09-26T16:17:14.000Z","ctime":"2016-09-22T17:46:21.000Z","inserttime":
"2018-07-24T16:32:59.000Z","tier":"system","size":929541048,"qpart":0,"fkey":
"filesys1.university.edufilesys13067112","project":null,"department":null,"backup":null,
"tag4":null,"tag5":null,"tag6":null,"tag7":null,"tag8":null,"tag9":null,"tag10":null,
"tag11":null,"tag12":null,"tag13":null,"tag14":null,"tag15":null,"tag16":null},
{"filesystem":"filesys1","operation":"INDEX","source":"SCAN","revision":"MO1","site":"",
"platform":"Spectrum Scale","cluster":"filesys1.university.edu","inode":3067092,"owner":
"root","group":"root","permissions":"-r--r--r--","fileset":"root","uid":9,"gid":10,
"path":"\/filesys1\/blast_db\/","filename":"other_genomic.65.tar.gz","filetype":"gz",
"migstatus":"resdnt","migloc":"NA","mtime":"2014-12-31T06:00:00.000Z","atime":
"2016-09-26T16:17:07.000Z","ctime":"2016-09-22T17:24:09.000Z","inserttime":
"2018-07-24T16:32:59.000Z","tier":"system","size":933683784,"qpart":0,"fkey":
"filesys1.university.edureflib3067092","project":null,"department":null,"backup":null,
"tag4":null,"tag5":null,"tag6":null,"tag7":null,"tag8":null,"tag9":null,"tag10":null,
"tag11":null,"tag12":null,"tag13":null,"tag14":null,"tag15":null,"tag16":null},
...
```


/db2whrest/v1/sql_query_async?<sql>: GET

Gets data from a database asynchronously with an SQL query in a GET command.

You can use any standard SQL query. You must include the SQL query as a parameter of the URL. To send the query in an HTTP message body, see /db2whrest/v1/sql_query_async: POST. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	√[1]	√[1]	X	X
[1]The search is restricted to documents that are tagged with collections to which the user ID has a datauser role assigned.				

The asynchronous endpoints are useful alternatives in situations where the operation takes a long time to complete or has a negative effect on the overall performance of the system. For more information, see the topic Asynchronous endpoints.

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/sql_query_async?<sql_statements> -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- CSV
- JSON

Examples

1. The following example shows how to get data from a database asynchronously with an SQL query in a GET command. The SQL query is included as a parameter of the URL.
   a. Submit the request. The endpoint responds by displaying the field `{"status": "work scheduled"}` to indicate that the operation is running asynchronously.

      ```
      curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/sql_query_async?
      select%20*%20from%20datasrv04%20where%20owner%3D%27root%27 -X GET {"status": "work scheduled"}
      ```

      The synchronous response to the request contains information about the asynchronously running task. The following is an example. For more information, see the topic Asynchronous endpoints.

      ```
      * HTTP 1.0, assume close after body
      < HTTP/1.0 202 ACCEPTED
      < Content-Type: text/html; charset=utf-8
      < Content-Length: 28
      < Location: https://data_cataloging_host/db2whrest/v1/task_status/25b2df40-8f0c-4e32-a2f9-999ff3b18ada
      < Server: Werkzeug/0.14.1 Python/3.4.5
      < Date: Thu, 15 Feb 2018 20:37:10 GMT
      ```

   b. Later, poll the URL that is provided in the `Location` field of the synchronous response header in Step (a).

      ```
      curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ task_status/25b2df40-8f0c-
      4e32-a2f9-999ff3b18ada
      ```

   c. When the operation is complete, the endpoint displays the rows of data that are returned in response to the SQL query in the request. In the following example only the first two rows of output are shown:

      ```
      0,reflib,INDEX,SCAN,MO1,,Spectrum Scale,reflib.university.edu,3067343,root,root,
      -rw-r--r--,root,9,10,/reflib/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,
      sjdbList.out.tab,tab,resdnt,NA,2016-11-14 19:40:10,2017-06-02 21:30:26,
      2017-06-02 21:30:26,2018-07-24 16:32:47,system,0,1,reflib.university.edureflib3067343,
      ,,,,,,,,,,,,,,,
      .......
      1,reflib,INDEX,SCAN,MO1,,Spectrum Scale,reflib.university.edu,3067333,root,root,
      -rw-r--r--,root,9,10,/reflib/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,
      chrLength.txt,txt,resdnt,NA,2016-11-14 19:40:10,2017-06-02 23:28:21,
      2017-06-02 21:30:26,2018-07-24 16:32:47,system,412,1,reflib.university.edureflib3067333,
      ,,,,,,,,,,,,,,,
      ...
      ```

/db2whrest/v1/sql_query_async: POST

Gets data from a database asynchronously with an SQL query in a POST command. You can use any standard SQL query. You must include the SQL query in the message body of the POST command.
To send the query as a parameter of the URL, see /db2whrest/v1/sql_query_async?<sql>: GET.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	√[1]	√[1]	X	X

[1]The search is restricted to documents that are tagged with collections to which the user ID has a datauser role assigned.

The asynchronous endpoints are useful alternatives in situations where the operation takes a long time to complete or has a negative effect on the overall performance of the system. For more information, see the topic Asynchronous endpoints.

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token> https://<data_cataloging_host>/db2whrest/v1/sql_query_async -X POST -d@<sql.dat>
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- CSV
- JSON

Examples

1. The following example shows how to get data from a database asynchronously with an SQL query in a POST command. The SQL query is sent in the POST message body. The output format is JSON.
   a. Write the record to be added to the database into a file named sql.dat.
   b. Submit the request:

      ```
      curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/sql_query_async -X POST -
      d"select * from metaocean where owner='root'" -H "accept: application/json"
      ```

      The synchronous response to the request contains information about the asynchronously running task. The following is an example. For more information, see the topic /db2whrest/v1/sql_query_async?<sql>: GET.

```
* HTTP 1.0, assume close after body
< HTTP/1.0 202 ACCEPTED
< Content-Type: text/html; charset=utf-8
< Content-Length: 28
< Location: https://<data_cataloging_host>/db2whrest/v1/task_status/3e40cfeb-efc9-4091-8bc8-
b194ff49d728
< Server: Werkzeug/0.14.1 Python/3.4.5
< Date: Thu, 15 Feb 2018 20:37:10 GMT
```

c. Later, poll the URL that is provided in the `Location` field of the synchronous response header in Step (b):

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/task_status/ 25b2df40-8f0c-
4e32-a2f9-999ff3b18ada
```

d. When the operation is complete, the endpoint displays the rows of data that are returned in response to the SQL query in the request. In the following example only the first two rows of output are shown. The output format is JSON:

```
[{"filesystem":"reflib","operation":"INDEX","source":"SCAN","revision":"MO1","site":"",
"platform":"Spectrum Scale","cluster":"reflib.university.edu","inode":3067112,"owner":
"root","group":"root","permissions":"-r--r--r--","fileset":"root","uid":9,"gid":10,
"path":"\/reflib\/blast_db\/","filename":"other_genomic.75.tar.gz","filetype":"gz",
"migstatus":"resdnt","migloc":"NA","mtime":"2014-12-31T06:00:00.000Z","atime":
"2016-09-26T16:17:14.000Z","ctime":"2016-09-22T17:46:21.000Z","inserttime":
"2018-07-24T16:32:59.000Z","tier":"system","size":929541048,"qpart":0,"fkey":
"reflib.university.edureflib3067112","project":null,"department":null,"backup":
null,"tag4":null,"tag5":null,"tag6":null,"tag7":null,"tag8":null,"tag9":null,"tag10":
null,"tag11":null,"tag12":null,"tag13":null,"tag14":null,"tag15":null,"tag16":null},

{"filesystem":"reflib","operation":"INDEX","source":"SCAN","revision":"MO1","site":"",
"platform":"Spectrum Scale","cluster":"reflib.university.edu","inode":3067092,"owner":
"root","group":"root","permissions":"-r--r--r--","fileset":"root","uid":9,"gid":10,
"path":"\/reflib\/blast_db\/","filename":"other_genomic.65.tar.gz","filetype":"gz",
"migstatus":"resdnt","migloc":"NA","mtime":"2014-12-31T06:00:00.000Z","atime":
"2016-09-26T16:17:07.000Z","ctime":"2016-09-22T17:24:09.000Z","inserttime":
"2018-07-24T16:32:59.000Z","tier":"system","size":933683784,"qpart":0,"fkey":
"reflib.university.edureflib3067092","project":null,"department":null,"backup":
null,"tag4":null,"tag5":null,"tag6":null,"tag7":null,"tag8":null,"tag9":null,"tag10":
null,"tag11":null,"tag12":null,"tag13":null,"tag14":null,"tag15":null,"tag16":null},
...
]
```

/db2whrest/v1/task_status/<task_id>: GET and <task_id>/peek

Gets the status of the specified asynchronous task.

The following table shows which roles can access these two REST API endpoints:

Table 1. Access by role

Endpoint	Data admin	Data user	Collection Admin	Admin	Service user
/db2whrest/v1/task_status/<task_id>: GET	√	√	√	X	X
/db2whrest/v1/task_status/<task_id>/peek: GET	√	√	√	X	X

The peek variation of this endpoint returns the status of the asynchronous operation. The nonpeek version returns both the status of the operation and a response header that includes further information. For more information about using these endpoints, see the following examples, and also see the topic Asynchronous endpoints.

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/task_status/ <task_id> -X GET
```

or

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/task_status/ <task_id>/peek -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- CSV
- JSON

Examples

1. In this example, the user discovers that the asynchronous operation completes and finds the location URI of the results of the operation. For more information on these steps, see the topic Asynchronous endpoints. The user runs the **`/db2whrest/v1/task_status/<task_id>: GET`** endpoint on the location URI to get the results of the operation:

   ```
   curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ task_status/9c0db090-bd33-41c5-
   969f-a1603ddf49ab
   ```

   The results are returned in the response. Only the first two rows of the results are shown in this example:

```
0,reflib,INDEX,SCAN,MO1,,Spectrum Scale,reflib.university.edu,3067343,root,root,
-rw-r--r--,root,9,10,/reflib/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,
sjdbList.out.tab,tab,resdnt,NA,2016-11-14 19:40:10,2017-06-02 21:30:26,2017-06-02
21:30:26,2018-07-24 16:32:47,system,0,1,reflib.university.edureflib3067343,
,,,,,,,,,,,,,,
1,reflib,INDEX,SCAN,MO1,,Spectrum Scale,reflib.university.edu,3067333,root,root,
-rw-r--r--,root,9,10,/reflib/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,
chrLength.txt,txt,resdnt,NA,2016-11-14 19:40:10,2017-06-02 23:28:21,2017-06-02 21:30:26,
2018-07-24 16:32:47,system,412,1,reflib.university.edureflib3067333,
,,,,,,,,,,,,,,
...
```

2. In this example, the user learns that the asynchronous operation is in progress and finds the location URI of the asynchronous operation. For more information on these tasks, see the topic Asynchronous endpoints. The user can do the following steps:
   a. Monitor the status of the asynchronous operation by running the **/db2whrest/v1/task_status/<*task_id*>/peek: GET** endpoint on the location URI of the asynchronous operation several times:

      ```
      curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/ task_status/9c0db090-bd33-
      41c5-969f-a1603ddf8888/peek
      ```

      In the first few tries the response to the peek endpoint indicates that the asynchronous operation is still running:

      ```
      {"status":work scheduled}
      ```

      On the final try the response to the peek endpoint indicates that the asynchronous operation completes:

      ```
      {"status":work scheduled}
      ```

   b. Find the location URI of the asynchronous operation. This location URI is already available in Step 2(a), where the peek endpoint is run on it.
   c. Get the location URI of the results of the operation. To accomplish this task, run the **/db2whrest/v1/task_status/<*task_id*>: GET** endpoint on the location URI of the asynchronous operation, which you obtained in Step 2(b). The response contains the location URI of the results of the operation.
   d. Get the results of the asynchronous operation. To accomplish this task, run the **/db2whrest/v1/task_status/<*task_id*>: GET** endpoint on the location URI of the results of the operation, which you obtained in Step 2(c):

      ```
      curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/task_status/9c0db090-bd33-
      41c5-969f-a1603ddf8888
      ```

      The results of the asynchronous operation are returned in the response. Only the first two rows of the results are shown in this example:

      ```
      0,reflib,INDEX,SCAN,MO1,,Spectrum Scale,reflib.university.edu,3067343,root,root,
      -rw-r--r--,root,9,10,/reflib/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,
      sjdbList.out.tab,tab,resdnt,NA,2016-11-14 19:40:10,2017-06-02 21:30:26,2017-06-02
      21:30:26,2018-07-24 16:32:47,system,0,1,reflib.university.edureflib3067343,
      ,,,,,,,,,,,,,,
      1,reflib,INDEX,SCAN,MO1,,Spectrum Scale,reflib.university.edu,3067333,root,root,
      -rw-r--r--,root,9,10,/reflib/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,
      chrLength.txt,txt,resdnt,NA,2016-11-14 19:40:10,2017-06-02 23:28:21,2017-06-02 21:30:26,
      2018-07-24 16:32:47,system,412,1,reflib.university.edureflib3067333,
      ,,,,,,,,,,,,,,
      ...
      ```

For more information, see the topic Asynchronous endpoints.

/db2whrest/v1/summary_tables -X GET

Gets information on all the summary tables that are available in the system.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	CollectionAdmin	Data user	Admin	Service user
√	X	X	X	X

Synopsis of the request URL

```
curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

1. The following example shows how to get information on all the summary tables that are available in the system.
   a. Submit the request:

      ```
      curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables -X GET
      ```

b. The following response is returned. The response data is in JSON format. The data has been manually reflowed to make it more readable.

```
{
  "duplicates": {
    "enabled": false,
    "last_duration": null,
    "last_updated_time": "2019-07-10T18:39:58.348Z",
    "schedule": {
      "hour": 0,
      "minute": "14"
    },
    "update_running": false,
    "update_start_time": null
  },
  "mrcapacity": {
    "enabled": true,
    "last_duration": "0:01:10",
    "last_updated_time": "2019-07-10T21:45:02.462Z",
    "schedule": {
      "minute": "*/5"
    },
    "update_running": true,
    "update_start_time": "2019-07-10_21:50:01"
  }
}
```

/db2whrest/v1/summary_tables/<table> -X GET

Gets information for a particular summary table available in the system.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	CollectionAdmin	Data user	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables/<table> -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

1. The following example shows how to get information on a particular summary table that is available in the system.
   a. Submit the request:

   ```
   curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables/mrcapacity
   ```

   b. The following response is returned. The response data is in JSON format. The data has been manually reflowed to make it more readable.

   ```
   {
     "enabled": true,
     "last_duration": "0:01:10",
     "last_updated_time": "2019-07-10T21:45:02.462Z",
     "schedule": {
       "minute": "*/5"
     },
     "update_running": true,
     "update_start_time":  "2019-07-10T22:00:00.342Z"
   }
   ```

/db2whrest/v1/summary_tables/<table> -X PUT

Change the summary table configuration for a particular summary table available in the system.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	CollectionAdmin	Data user	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables/<table> -X PUT -d@data.json -H "Content-Type:
application/json"
```

Supported request types and response formats

Supported request types:

- PUT

Supported response formats:

- JSON

Examples

1. The following example shows how to disable auto scheduling for a particular summary table that is available in the system.
   a. Submit the request:

   ```
   curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables/mrcapacity -X PUT -d'{"enabled": false}' -H
   "Content-Type: application/json"
   ```

   b. The following response is returned.

   ```
   {"status": "success"}
   ```

2. The following example shows how to enable auto scheduling for a particular summary table that is available in the system.
   a. Submit the request:

   ```
   curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables/mrcapacity -X PUT -d'{"enabled": true,
   "schedule": {"minute": "*/5"}}' -H "Content-Type: application/json"
   ```

   b. The following response is returned.

   ```
   {"status": "success"}
   ```

3. The following example shows how to change the configuration schedule for a particular summary table that is available in the system.
   a. Submit the request:

   ```
   curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables/mrcapacity -X PUT -d'{"enabled": true,
   "schedule": {"minute": "*/5"}}' -H "Content-Type: application/json"
   ```

   b. The following response is returned.

   ```
   {"status": "success"}
   ```

/db2whrest/v1/summary_tables/<table>/<action> -X PUT

Triggers a start or scheduled start action for a particular summary table.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	CollectionAdmin	Data user	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables/<table>/<action> -X PUT
```

Supported request types and response formats

Supported request types:

- PUT

Supported response formats:

- JSON

Examples

1. The following example shows how to start a table refresh on demand for a particular summary table that is available in the system.
   a. Submit the request:

   ```
   curl -u <user>:<pass> https://<master_node>/db2whrest/v1/summary_tables/mrcapacity/start -X PUT
   ```

b. The following response is returned.

```
  {"status": "success"}
```

/db2whrest/v1/bulk_add_tags/docs: POST

Searches a database for data.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✔	✔[1]	✔[1]	X	X
[1]The search is restricted to documents that are tagged with collections to which the user ID has a datauser role assigned.				

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/bulk_add_tags/docs -X POST -
d@bulk_of_docs.json -H "Content-type: application/json"
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

1. The following example shows how to define search parameters and format the data that is returned:
   a. Step 1: Define the search parameters in a file named bulk_of_docs.json:

```
{
  "docs": {
      "fkey1": {
          "tags": {
            "tag1": "This is the value for tag1",
            "tag2": "This is the value for tag2",
            "tag3": "This is the value for tag3",
          }
      },
      "fkey2": {
          "tags": {
            "tag4": "This is the value for tag4",
            "tag5": "This is the value for tag5",
            "tag6": "This is the value for tag6",
          }
      },
  }
}
```

   b. Step 2: Submit the request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/db2whrest/v1/bulk_add_tags/docs -X POST -
d@bulk_of_docs.json -H "Content-Type: application/json"
```

   The following response is returned:

```
(200 OK)
{"status": "success"}
```

Endpoints for working with policy management

With the policy management API service, you can create, list, update, and delete the policies.

A policy API consists of the following elements:

pol_id
: The name of the policy.

action_id
: The action to perform.

action_params
: The action contains the following elements:

  tags

The type of actions. Currently, the only type of action is the AUTOTAG action. For information about using the AUTOTAG action, see the topic Adding fields with the AUTOTAG action.

pol_filter
: A search filter to select a set of documents to work on.

schedule
: The schedule for processing the operation:

  NOW
  : The action is done immediately.

  dayOfWeek, hour, month, timezone, minute, dayOfMonth
  : The action is done at the specified time.
  Note: The API supports the creation of a schedule by using a timezone parameter. Internally, this information is converted to Coordinated Universal Time (UTC). It is recommended that the timezone parameter is not used or that the default of Coordinated Universal Time (UTC) is used in the API, because the policy schedule time is not displayed in the GUI. The GUI assumes that all times are in Coordinated Universal Time (UTC).

The policy engine runs the specified policy based on the schedule and the action ID that are specified in the API. The policy operations are applied to the documents that are specified by the filter.

- **/policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET**
  Lists the attributes of one or more policies.
- **/policyengine/v1/policies/<policy_name>/preview: GET**
  Previews a policy.
- **/policyengine/v1/policies/<policy_name>/status:GET**
  Gets policy status.
- **/policyengine/v1/policyhistory: GET**
  Retrieves policy history information for all the policies.
- **/policyengine/v1/policyhistory/<pol_id>/<log_id>: GET**
  Retrieves the policy run log for a policy execution.
- **/policyengine/v1/policies/<policy_name>/<action>: POST**
  Applies action to policy.
- **/policyengine/v1/policies -d '<data>': POST**
  Creates a policy.
- **/policyengine/v1/policies/<policy_name> -d '<data>': PUT**
  Updates a policy.
- **/policyengine/v1/policies/<policy_name>: DELETE**
  Deletes a policy.
- **/policyengine/v1/policyhistory: DELETE**
  Deletes the policy execution history and logs.
- **/policyengine/v1/tags: GET**
  Retrieves tag definitions.
- **/policyengine/v1/tags/: POST**
  Updates tags definitions.
- **/policyengine/v1/tags/: PUT**
  Adds one or more new tags to the list of existing definitions.
- **/policyengine/v1/tags/: DELETE**
  Deletes an existing tag definition.
- **/policyengine/v1/regex: POST**
  Registers the regular expression in the IBM Data Cataloging system.
- **/policyengine/v1/regex: GET**
  Retrieves list of all the regular expressions registered in the IBM Data Cataloging system.
- **/policyengine/v1/regex/<regex_name>: GET**
  Retrieves a single regular expression detail from the IBM Data Cataloging system.
- **/policyengine/v1/regex: PUT**
  Updates the regular expressions in the IBM Data Cataloging system.
- **/policyengine/v1/regex: DELETE**
  Deletes the regular expression with value from the IBM Data Cataloging system.
- **Adding fields with the AUTOTAG action**
  With the AUTOTAG action you can create policies that automatically add fields either to a specified set of documents or to all the documents in the system.

/policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET

Lists the attributes of one or more policies.

These two endpoints list the attributes either of a specified policy or of all the policies in the system. The following table shows which roles can access these two REST API endpoints:

Table 1. Access by role

Endpoints	Data admin	Data user	Collection Admin	Collection user	Admin	Service user
/policyengine/v1/policies: GET	✓	✓	✓[1]	✓	X	X
/policyengine/v1/policies/<policy_name>: GET	✓	✓	✓[1]	✓	X	X
[1] Collection Admin user can list, update, and delete policies applied to the collections to which they have the Collection Admin role assigned.						

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/ policies -X GET -H 'Accept: application/json'
```

or

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/ policies/<policy_name> -X GET -H
'Accept: application/json'
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

1. The following example returns information about all the policies that are configured in the system:
   a. Issue the following request on one line:

   ```
   curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host> /policyengine/v1/policies -X GET -H
   'Accept: application/json'
   ```

   b. The following response is returned:

   ```
   [
      {
         "pol_id": "pol1",
         "action_id": "AUTOTAG",
         "action_params": {
            "tags": {
            "tag2": "val2",
            "tag3": "val3"
            }
         },
         "schedule": {
         "dayOfWeek": 4,
         "hour": 5,
         "minute": 15
         }
         "pol_state": "active"
         "pol_filter": "filetype='jpg'",
      },
   {
      "pol_id": "pol2",
         "action_id": "AUTOTAG",
         "action_params": {
            "tags": {
            "tag4": "val4",
            "tag5": "val5"
            }
         },
         "schedule": "NOW",
         "pol_state": "active"
         "pol_filter": "filetype='jpg'",
      }
   ]
   ```

2. The following example returns information about the specified policy:
   a. Issue the following request on one line:

   ```
   curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host> /policyengine/v1/policies/pol1 -X GET -H
   'Accept: application/json'
   ```

   b. The following response is returned:

   ```
   [
      {
         "pol_id": "pol1",
         "action_id": "AUTOTAG",
         "action_params": {
            "tags": {
            "tag2": "val2",
            "tag3": "val3"
            }
         },
         "schedule": {
         "dayOfWeek": 4,
         "hour": 5,
         "minute": 15
         }
         "pol_filter": "filetype='jpg'",
         "pol_state": "active"
      }
   ]
   ```

/policyengine/v1/policies/<policy_name>/preview: GET

Previews a policy.

This endpoint helps to display policy preview information for a specific policy. The following information is available for preview:

Estimation information
: Displays the total count and size of documents. The size of documents on the disk is calculated based on the list of documents the policy applies to and those that are returned.

Execution information
: Displays whether the policy has been previously executed or is currently running. The information returned also includes the number and size of documents completed, the start time and, the duration of the policy in seconds.

Policy schedule
: Displays the policy schedule, if any.

Policy run time
: Displays the future run date and time that has been scheduled for the policy.

The following table displays the roles that can access the REST API endpoint.

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
✓	✓	✓	✓	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host> /policyengine/v1/policies/policy1/preview -X GET -H
'Accept: application/json'
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example returns the policy preview information about a specific policy that is configured in the system.

Issue the following request in one line:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/ policies/<policy_name>/preview -X
GET -H 'Accept: application/json'
```

The following response is returned:

```
{
  "schedule": {
    "minute": "18",
    "hour": "12",
    "dayOfMonth": "*",
    "month": "*",
    "dayOfWeek": "*",
    "timezone": "UTC"
  },
  "next_run_time": "2020-03-19 12:18PM",
  "estimation_info" : {
    "document_count": 152467,
    "total_size": 24935050549,
    "total_size_on_disk": 25610976130
   },
  "execution_info": {
    "start_time": "",
    "duration": 3000,
    "tier_count": 10345,
   "tier_size": 1000000,
   "failed_count": 0
  }
}
```

/policyengine/v1/policies/<policy_name>/status:GET

Gets policy status.

The status endpoint retrieves the policy status information for a specific policy.

The response includes the following information:

- total_count
- completed_count

- failed_count
- submitted_count
- start_time
- end_time

The response returned depends on three possible policy scenarios. These include Never Run, Running, and Finished. If the policy is never executed (Never Run), then the response only returns the policy status.

If the policy is running, the counts mentioned in the preceding list contain the current values and an end time is not returned.

If the policy is finished then the counts contain the final values and an end time is returned.

The following table displays the roles that can access the REST API endpoint.

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
✓	[1]✓	[1]✓	✓	X	X
[1] Users with Collection Admin role can access this endpoint only if they are owners of the policy or can access one or more collections that are included in the policy. Users with Data User role can access the endpoint only if they are owners of the policy.					

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/ policies/<policy_name>/status -X GET -H 'Accept: application/json'
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example returns policy status information about a specific policy.

Issue the following request in one line:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host> /policyengine/v1/policies/policy1/status -X GET -H 'Accept: application/json'
```

The following response is returned:

```
{
    "status": "running",
    "total_count": 15000,
    "completed_count": 14000,
    "failed_count": 0,
    "submitted_count": 15000,
    "start_time": "2020-03-27_11:12:10"
}
```

/policyengine/v1/policyhistory: GET

Retrieves policy history information for all the policies.

The /policyengine/v1/policyhistory retrieves the policy execution history details for all the policies that you have permission to view within the system. The entries contain detailed information on the specified policy execution. It also provides the location of the run log file.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
✓	✓	✓	✓	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/policyhistory -X GET -H 'Accept: application/json'
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example returns the policy history information for all policies for which you have view permissions.

1. Issue the following request in one line.

```
curl -k -H 'Authorization: Bearer ${TOKEN}' https://<data_cataloging_host>/policyengine/v1/policyhistory -X GET
```

2. The following response is returned.


```
[
  {
    "id": 893909178,
    "policyid": "patent_pol",
    "state": "complete",
    "status": null,
    "starttime": "2020-09-22T20:34:32.000Z",
    "endtime": "2020-09-22T20:35:19.000Z",
    "logfile": "/policy_logs/patent_pol-2020-09-22 20:34:32/run.log",
    "startedby": null,
    "scheduled": null,
    "totalcount": 7,
    "failedcount": 0,
    "skippedcount": 0,
    "successcount": 7,
    "totalsize": null,
    "policyfilter": "path = '/mnt/datadump/patent_dataset/'",
    "type": "CONTENTSEARCH",
    "additional_info": "{\"explicit\": \"true\"}"
  },
  {
    "id": 3465755378,
    "policyid": "dicom_hdr_extract",
    "state": "complete with fail",
    "status": null,
    "starttime": "2020-09-22T21:30:59.000Z",
    "endtime": "2020-09-22T21:31:09.000Z",
    "logfile": "/policy_logs/dicom_hdr_extract-2020-09-22 21:30:59/run.log",
    "startedby": null,
    "scheduled": null,
    "totalcount": 26,
    "failedcount": 9,
    "skippedcount": 0,
    "successcount": 17,
    "totalsize": null,
    "policyfilter": "project='watson_health' and filetype='dcm'",
    "type": "CONTENTSEARCH",
    "additional_info": "{\"explicit\": \"true\"}"
  }
]
```


Response data

The response data can be explained as shown.

id
: The response returns the unique run id for the specified policy execution.

policy_id
: The response returns the name of the policy that is being run.

state
: The response returns the current or final state of the policy execution.

status
: The response returns the additional information about the run state (for example, a failure message).

starttime
: The response returns the date or timestamp when the policy execution was started.

endtime
: The response returns the date or timestamp when the policy execution completes if the policy is not still running.

logfile
: The response returns the log file identifier for this policy execution.

startedby
: This information field is reserved for future use.

scheduled
: This information field is reserved for future use.

totalcount
: The response returns the total number of records that meet the policy filter criteria.

failedcount
: The response returns the number of records that are marked as failed during the policy execution.

skippedcount
: The response returns the number of records that are marked as skipped during the policy execution.

successcount
: The response returns the number of records that are marked as successfully processed after the policy execution is completed.

totalsize
: This information field is reserved for future use.

policyfilter

The response returns the filter criteria that defined the set of records to be processed in the policy execution.

type
: The response returns the type of policy that is being run.

additional_info/explicit
: The response returns the Boolean value that displays *True* if the policy execution was for a user-defined policy or *False* if it was run for a system-generated policy.

/policyengine/v1/policyhistory/<pol_id>/<log_id>: GET

Retrieves the policy run log for a policy execution.

This endpoint helps you retrieve and view the policy run log information for all policy executions that you have access to. It provides you with a direct access to the policy execution log data without having to search for it on the file system.

The following table displays the roles that can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
✓	✓	✓	✓	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/policyhistory/<pol_id>/<log_id> -X
GET -H 'Accept: application/json'
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example returns the policy log execution details for the policies that you have access to:

1. Issue the following request in one line.

```
curl -k -H 'Authorization: Bearer ${TOKEN}'
https://<data_cataloging_host>/policyengine/v1/policyhistory/patent_pol/893909178 -X GET -H 'Accept: application/json'
```

2. The following response is returned.


```
[
  {
  "log": "[2020-09-22 20:34:32] - Execution beginning for policy (patent_pol)\n[2020-09-22 20:34:33] - Policy status
change to (running)\n[2020-09-22 20:34:33] - Policy stats update: {'pol_id': 'patent_pol', 'execution_info':
'{\"start_time\": \"2020-09-22 20:34:32\", \"total_count\": 0, \"submitted_count\": 0, \"failed_count\": 0,
\"completed_count\": 0, \"skipped_count\": 0, \"ctrl_submitted_count\": 0, \"ctrl_completed_count\": 0,
\"contentsearch_count\": 0, \"run_id\": null}'}\n[2020-09-22 20:34:33] - Policy stats update: {'pol_id': 'patent_pol',
'execution_info': '{\"start_time\": \"2020-09-22 20:34:32\", \"total_count\": 7, \"submitted_count\": 0, \"failed_count\":
0, \"completed_count\": 0, \"skipped_count\": 0, \"ctrl_submitted_count\": 0, \"ctrl_completed_count\": 0,
\"contentsearch_count\": 0, \"run_id\": null}'}\n[2020-09-22 20:34:35] - Applying action 'CONTENTSEARCH' to 7
documents\n[2020-09-22 20:34:38] - Policy stats update: {'pol_id': 'patent_pol', 'execution_info': '{\"start_time\":
\"2020-09-22 20:34:32\", \"total_count\": 7, \"submitted_count\": 7, \"failed_count\": 0, \"completed_count\": 0,
\"skipped_count\": 0, \"ctrl_submitted_count\": 0, \"ctrl_completed_count\": 0, \"contentsearch_count\": 0, \"run_id\":
\"ab0d0623dd844fd7b21276fce892d6ed\"}'}\n[2020-09-22 20:34:38] - Policy stats update: {'pol_id': 'patent_pol',
'execution_info': '{\"start_time\": \"2020-09-22 20:34:32\", \"total_count\": 7, \"submitted_count\": 7, \"failed_count\":
0, \"completed_count\": 0, \"skipped_count\": 0, \"ctrl_submitted_count\": 0, \"ctrl_completed_count\": 0,
\"contentsearch_count\": 0, \"run_id\": \"ab0d0623dd844fd7b21276fce892d6ed\"}'}\n[2020-09-22 20:34:45] - Policy stats
update: {'pol_id': 'patent_pol', 'execution_info': '{\"start_time\": \"2020-09-22 20:34:32\", \"total_count\": 7,
\"submitted_count\": 7, \"failed_count\": 0, \"completed_count\": 0, \"skipped_count\": 0, \"ctrl_submitted_count\": 0,
\"ctrl_completed_count\": 0, \"contentsearch_count\": 0, \"run_id\": \"ab0d0623dd844fd7b21276fce892d6ed\"}'}\n[2020-09-22
20:34:56] - Policy stats update: {'pol_id': 'patent_pol', 'execution_info': '{\"start_time\": \"2020-09-22 20:34:32\",
\"total_count\": 7, \"submitted_count\": 7, \"failed_count\": 0, \"completed_count\": 0, \"skipped_count\": 0,
\"ctrl_submitted_count\": 0, \"ctrl_completed_count\": 0, \"contentsearch_count\": 0, \"run_id\":
\"ab0d0623dd844fd7b21276fce892d6ed\"}'}\n[2020-09-22 20:35:06] - Policy stats update: {'pol_id': 'patent_pol',
'execution_info': '{\"start_time\": \"2020-09-22 20:34:32\", \"total_count\": 7, \"submitted_count\": 7, \"failed_count\":
0, \"completed_count\": 0, \"skipped_count\": 0, \"ctrl_submitted_count\": 0, \"ctrl_completed_count\": 0,
\"contentsearch_count\": 0, \"run_id\": \"ab0d0623dd844fd7b21276fce892d6ed\"}'}\n[2020-09-22 20:35:16] - Policy stats
update: {'pol_id': 'patent_pol', 'execution_info': '{\"start_time\": \"2020-09-22 20:34:32\", \"total_count\": 7,
\"submitted_count\": 7, \"failed_count\": 0, \"completed_count\": 0, \"skipped_count\": 0, \"ctrl_submitted_count\": 0,
\"ctrl_completed_count\": 0, \"contentsearch_count\": 0, \"run_id\": \"ab0d0623dd844fd7b21276fce892d6ed\"}'}\n[2020-09-22
20:35:19] - Policy stats update: {'pol_id': 'patent_pol', 'execution_info': '{\"start_time\": \"2020-09-22 20:34:32\",
\"total_count\": 7, \"submitted_count\": 7, \"failed_count\": 0, \"completed_count\": 7, \"skipped_count\": 0,
\"ctrl_submitted_count\": 0, \"ctrl_completed_count\": 0, \"contentsearch_count\": 0, \"run_id\":
\"ab0d0623dd844fd7b21276fce892d6ed\"}'}\n[2020-09-22 20:35:20] - Policy stats update: {'pol_id': 'patent_pol',
'execution_info': '{\"start_time\": \"2020-09-22 20:34:32\", \"total_count\": 7, \"submitted_count\": 7, \"failed_count\":
0, \"completed_count\": 7, \"skipped_count\": 0, \"ctrl_submitted_count\": 0, \"ctrl_completed_count\": 0,
\"contentsearch_count\": 0, \"run_id\": \"ab0d0623dd844fd7b21276fce892d6ed\", \"end_time\": \"2020-09-22
20:35:19\"}'}\n[2020-09-22 20:35:20] - Policy patent_pol run ending\n[2020-09-22 20:35:20] - Policy patent_pol run
completed\n[2020-09-22 20:35:20] - Policy status change to (complete)\n"
}
```

```
]
```

Response data

The response data can be explained as shown.

log
: The response returns the policy run log output details that is displayed as a string.

/policyengine/v1/policies/<policy_name>/<action>: POST

Applies action to policy.

This API endpoint helps you start, kill, pause, and resume a policy. You need to add the required action start, pause, resume, kill to the following path: /policyengine/v1/policies/{pol_id}/{action}.

The table shows the roles that can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
√	[1]√	[1]√	X	X	X
[1] Users with Collection Admin role can access this endpoint only if they are owners of the policy or can access one or more collections that are included in the policy. Users with Data User role can access this endpoint only if they are owners of the policy.					

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/policies/policy_name/action -X POST -H 'Content-type: application/json'
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

The following example shows how to start a policy by applying the **start** action.

Issue the following request on one line:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/pol_2/start -X POST -H 'Content-type: application/json'
```

The following response is returned:

```
Accepted
```

/policyengine/v1/policies -d '<data>': POST

Creates a policy.

The **/policyengine/v1/policies: POST** endpoint creates a policy with the characteristics that are specified in the request data. For a description of the request parameters, see Endpoints for working with policy management. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Collection Admin	Collection user	Data user	Admin	Service user
√	√	X	√[1, 2]	X	X
[1]The command allows access only if the schedule attribute is set to NOW. [2]The command applies the policy only in collections where the user has a datauser role.					

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/policies -d '<data>' -X POST -H "Content-Type: application/json"
```

Supported request types, input fields, and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

1. The following example creates an AUTOTAG policy named `pol2` to start immediately:
   a. Issue the following request on one line:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/policyengine/v1/policies
-d '{
   "pol_id":"pol2",
   "pol_filter": "user='research1'",
   "action_id": "AUTOTAG",
   "action_params": {"tags": {"tag4":"val4", "tag5":"val5"}},
   "schedule":"NOW"
}'
 -X POST -H "Content-Type: application/json"
```

   b. The following response is returned:

```
Policy 'pol2' added
```

2. The following example creates a AUTOTAG policy named `pol2` to start at a scheduled time:
   Note: For information about using the AUTOTAG action, see the topic Adding fields with the AUTOTAG action.
   a. Issue the following request on one line:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/policyengine/v1/policies
-d '
{
  "pol_id":"pol2",
  "action_id":"AUTOTAG",
  "action_params":{
      "tags": { "test_tag1": "test1" }
    },
  "pol_filter":"filename LIKE 'f0%'",
  "schedule": {
      "dayOfWeek": "*",
      "hour": "13",
      "month": "*",
      "timezone": "UTC",
      "minute": "31",
      "dayOfMonth": "*"
    }
}'
 -X POST -H "Content-Type: application/json"
```

   Note: The API supports the creation of a schedule using a **timezone** parameter. Internally this is converted to Coordinated Universal Time (UTC). It is recommended that the **timezone** parameter is not used or that the default of Coordinated Universal Time (UTC) is used in the API, because the policy schedule time is not displayed in the GUI. The GUI assumes all times are in UTC.
   b. The following response is returned:

```
Policy 'pol2' added
```

To verify that the policy is created, issue a GET request to list the information about the specified policy or about all the policies in the system. For more information, see /policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET.

/policyengine/v1/policies/<policy_name> -d '<data>': PUT

Updates a policy.

The `/policyengine/v1/policies/<policy_id> -d '<data>': PUT` endpoint updates an existing policy with the attribute values that are specified in the request data. Attributes that are not updated keep the same values that they had before the update. The following attributes can be updated: `action_parameters`, `schedule`, `pol_filter`, and `pol_state`. The `action_id` attribute cannot be updated. However, you can delete an existing policy and then create a new one with the same name and attributes as the deleted policy but with a different `action_id`. For more information about the attributes in the request data, see Endpoints for working with policy management.

Note: You cannot update a policy while it is running.
The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
√	√[1, 2]	X	X	X	X
[1]The command allows access only if the schedule attribute is set to NOW. [2]The command applies the policy only in collections where the user has a datauser role.					

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/policies/<policy_name> -d '<data>' -X
PUT -H "Content-Type: application/json"
```

Supported request types and response formats

Supported request types:

- PUT

Supported response formats:

- JSON

Examples

1. The following example updates a policy named pol2:
   a. Issue the following request on one line:

```
curl -k -H 'Authorization: Bearer \<token>'
https://<data_cataloging_host>
/policyengine/v1/policies/<policy_name>
-d '{
   "pol_filter": "user='research1'",
   "pol_state": "active"
   "action_params": {"tags": {"tag4":"val4", "tag5":"val5"}},
   "schedule":"NOW"
}'
 -X POST -H "Content-Type: application/json"
```

   b. The following response is returned:

```
Policy 'pol2' updated
```

To verify that the policy is created, issue a GET request to list the information about the specified policy or about all the policies in the system. For more information, see /policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET.

/policyengine/v1/policies/<policy_name>: DELETE

Deletes a policy.

The **/policyengine/v1/policies/<*policy_name*>: DELETE** endpoint deletes the specified policy. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	√	X	X	X

Important: You cannot delete the **"Collection"** and **"Temperature"** tags.

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/policies/pol2 -X DELETE
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

1. The following example deletes a policy named pol2:
   a. Issue the following request on one line:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/policies/<policy_name> -X
POST
```

   b. The following response is returned:

```
Deleted policy pol2
```

To verify that the policy is deleted, issue a GET request to list the information about the specified policy or about all the policies in the system. For more information, see /policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET.

/policyengine/v1/policyhistory: DELETE

Deletes the policy execution history and logs.

This endpoint helps you delete the policy execution history and logs of all policies that you can access.

The following table displays the roles that can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
✓	✓	✓	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/policyengine/v1/policyhistory -X DELETE -H 'Accept: application/json' [-d '{"days_older_than": <int>}']
```

Note: You can specify an override to the default retention of 30 days in the JSON input data field, `{"days_older_than": <int>}`. When specified, it ensures that only the policy history records and log files that are older than the input value provided, are removed.

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

The following example deletes the policy history entries and run logs, for any policy executions that you have permission to view and that are older than 30 days.

1. Issue the following request in one line.

   ```
   curl -k -H 'Authorization: Bearer ${TOKEN}' https://<data_cataloging_host>/policyengine/v1/policyhistory -X DELETE -H 'Accept: application/json'
   ```

2. The following response is returned.

   ```
   [
     {"status": "Success. Deleted 12 policy history records."}
   ]
   ```

Response data

The response data can be explained as shown.

status
: The response returns an indication whether the request succeeded or failed. For a successful request, this information field also indicates the number of policy execution history entries and the corresponding log files that were purged.

/policyengine/v1/tags: GET

Retrieves tag definitions.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
✓	✓	✓	✓	X	X

Synopsis of the request URL

```
$ curl -H "Authorization: Bearer <token>" -k https://<data_cataloging_host>:443/policyengine/v1/tags
```

Supported request types and response formats

Supported request types:

- GET

Note: When a request type is not specified, curl defaults to a GET operation.
Supported response formats:

- JSON

Examples

The following response is returned:

```
[
  {
    "tag": "project",
    "type": "Open",
    "value": "[]"
  },
  {
    "tag": "chargeBack",
    "type": "Restricted",
    "value": "[\"dept1\",\"dept2\",\"dept3\"]"
  },
  {
    "tag": "classification",
    "type": "Restricted",
    "value": "[\"public\",\"confidential\",\"internal\",\"restricted\",\"other\"]"
  },
  {
    "tag": "demotag",
    "type": "Open",
    "value": "[]"
  },
  {
    "tag": "instrument",
    "type": "Restricted",
    "value": "[\"spectrometer\",\"microscopeA\",\"microscopeB\"]"
  }
]
```

/policyengine/v1/tags/: POST

Updates tags definitions.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
√[1]	√[2]	√	X	X	X
[1] Has read/write access to `'Open'`, `'Restricted'`, and `'Characteristics'` types of tags. [2] Has read-only access to `'Open'` and `'Restricted'` types of tags, and read/write access to `'Characteristics'` tags.					

To create an `'Open'` tag, run this command:

```
$ curl -k -H "Authorization: Bearer <token>" https://<data_cataloging_host>:443/policyengine/v1/tags -d '{"tag": "project",
"type": "Open"}' -X POST -H "Content-Type: application/json"
```

Synopsis of the request URL

```
$ curl -k -H "Authorization: Bearer <token>"
https://<data_cataloging_host>:443/policyengine/v1/tags -d '{"tag": "department",
"type": "Restricted", "value": ["Finance", "HR", "IT"]' -X POST -H "Content-Type:
application/json"
```

```
$ curl -k -H "Authorization: Bearer <token>"
https://<data_cataloging_host>:443/policyengine/v1/tags -d '{"tag": "annotations",
"type": "Characteristics"}' -X POST -H "Content-Type: application/json"
```

Supported request types and response formats

Supported request types:

- POST
- Specify one of the following tag `"type"` options:

  `"Open"` | `"Restricted"` | `"Characteristics"`

Note:

- `"Open"` allows the unrestricted setting of tag values with a maximum string length of 256 characters.
- `"Restricted"` allows the definition of specified tag values with a maximum string length of 256 characters.
- `"Characteristics"` allows the definition of unspecified tag values with a maximum string length of 4096 characters.

Supported response formats:

- JSON

/policyengine/v1/tags/: PUT

Adds one or more new tags to the list of existing definitions.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection admin	Collection user	Admin	Service user
√[1]	√[2]	√	X	X	X
[1]READ/WRITE access to 'Open', 'Restricted', and 'Characteristics' types of tags. [2]READ Only access to 'Open' and 'Restricted' types of tags, READ/WRITE access to 'Characteristics' tags.					

Synopsis of the request URL

```
$ curl -H "Authorization: Bearer <token>" -k
https://<data_cataloging_host>:443/policyengine/v1/tags/<tag_name> -d '{"value":
["dept1", "dept2", "dept3", "dept4"]}' -X PUT -H "Content-Type: application/json"
```

Supported request types and response formats

Supported request types:

- PUT

Supported response formats:

- JSON

Note: Tag `payload` update requests are applied to the existing tag. Update tag requests overwrite the existing attributes of a tag. However, you cannot change the tag type.

/policyengine/v1/tags/: DELETE

Deletes an existing tag definition.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Collection user	Admin	Service user
√[1]	X[2]	X	X	X	X
[1] For read/write access to `'Open'`, `'Restricted'`, and `'Characteristics'` types of tags. [2] For read-only access to `'Open'` and `'Restricted'` types of tags, and read/write access to `'Characteristics'` tags.					

Important: You cannot delete the `"Collection"` and `"Temperature"` tags.

Synopsis of the request URL

```
$ curl -H "Authorization: Bearer <token>" -k https://<data_cataloging_host>:443/policyengine/v1/tags/<tag_name> -X DELETE
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

1. If a tag specified from the set of currently defined tags is "in-use" (its nondefault value is set in at least one document) the delete tag operation returns an error:

   ```
   < HTTP/1.0 403 FORBIDDEN
   < Content-Type: text/html; charset=utf-8
   < Content-Length: 102
   < Server: Werkzeug/0.14.1 Python/2.7.5
   < Date: Wed, 09 May 2018 17:20:05 GMT
   <
   Cannot delete tag <tag_name> because it is in use in '38' records. Force deletion with
   'force' option.
   ```

2. To force deletion of an in-use tag, use the `'force'` option as follows:

   ```
   $ curl -H "Authorization: Bearer <token>" -k
   https://<data_cataloging_host>:443/policyengine/v1/tags/<tag_name> -X DELETE -d
   '{"force":"true"}' -H "Content-Type: application/json"
   ```

   Important: The force option results in the deletion of the tag and its values are set by default to `NULL` or empty string.

/policyengine/v1/regex: POST

Registers the regular expression in the IBM Data Cataloging system.

The table shows the roles that can perform create operation on the endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓	✓	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>'
https://$SDHOST/policyengine/v1/regex -d @regexp.json -H "Content-type:application/json" -X POST
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

The following example shows how to register a regular expression.

Issue the following request on one line:

```
curl -k -H "Authorization: Bearer ${TOKEN}" https://$SDHOST/policyengine/v1/regex -d @regexp.json -H "Content-
type:application/json" -X POST
```

Example: Request JSON file

```
{
  "regex_id": "US-SSN-blank-delimited",
  "pattern": "\\b\\d{3}\\s\\d{2}\\s\\d{4}\\b",
  "description": "US SSN delimited by white space, not dashes."
}
```

The following response is returned:

```
201(Created)
```

/policyengine/v1/regex: GET

Retrieves list of all the regular expressions registered in the IBM Data Cataloging system.

The following table shows roles that can perform read operation on the endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓	✓	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://$SDHOST/policyengine/v1/regex -X GET -H 'Accept: application/json'
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example returns all the regular expressions registered for which you have read permission.

1. Issue the following request in one line.

   ```
   curl -k -H "Authorization: Bearer ${TOKEN}" https://$SDHOST/policyengine/v1/regex -X GET
   ```

2. The following response is returned.

   ```
   200(OK)

   [
   ```

```
  {
    "regex_id": "EmailID",
    "pattern": "\\b[\\w\\.=-]+@[\\w\\.-]+\\.[\\w]{2,3}\\b",
    "description": "Matching Email IDs like : John.Smith@example.com"
  },
  {
    "regex_id": "US-SSN",
    "pattern": "\\b\\d{3}-\\d{2}-\\d{4}\\b",
    "description": "Matching United States Social Security Numbers (SSN) like: 513-84-7329"
  },
  {
    "regex_id": "IPV4-Address",
    "pattern": "\\b\\d{1,3}[.]\\d{1,3}[.]\\d{1,3}[.]\\d{1,3}\\b",
    "description": "Matching IPV4 address like: 192.168.1.1"
  },
  {
    "regex_id": "Dates-MM/DD/YYYY",
    "pattern": "\\b(((0)[0-9])|((1)[0-2]))(\\/)([0-2][0-9]|(3)[0-1])(\\/)\\d{4}\\b",
    "description": "Matching dates in MM/DD/YYYY format like: 05/21/2019"
  },
  {
    "regex_id": "Dates-DD/MM/YYYY",
    "pattern": "\\b([0-2][0-9]|(3)[0-1])(\\/)(((0)[0-9])|((1)[0-2]))(\\/)\\d{4}\\b",
    "description": "Matching dates in DD/MM/YYYY format like: 15/10/2019"
  },
  {
    "regex_id": "MasterCard",
    "pattern": "\\b(?:5[1-5][0-9]{2}|222[1-9]|22[3-9][0-9]|2[3-6][0-9]{2}|27[01]
               [0-9]|2720)[0-9]{12}\\b",
    "description": "Matching MasterCard number like: 5258704108753590"
  },
  {
    "regex_id": "VisaCard",
    "pattern": "\\b([4]\\d{3}[\\s]\\d{4}[\\s]\\d{4}[\\s]\\d{4}|[4]\\d{3}[-]\\d{4}[-]\\d{4}[-]
               \\d{4}|[4]\\d{3}[.]\\d{4}[.]\\d{4}[.]\\d{4}|[4]\\d{3}\\d{4}\\d{4}\\d{4})\\b",
    "description": "Matching Visa Card numbers like: 4563-7568-5698-4587"
  },
  {
    "regex_id": "AmexCard",
    "pattern": "\\b3[47][0-9]{13}\\b",
    "description": "Matching American Express Card numbers like: 340000000000009"
  },
  {
    "regex_id": "USZIPCode",
    "pattern": "\\b((\\d{5}-\\d{4})|(\\d{5})|([A-Z]\\d[A-Z]\\s\\d[A-Z]\\d))\\b",
    "description": "Matching United States ZIP codes like: 97589"
  },
  {
    "regex_id": "URL",
    "pattern": "\\b((http[s]?|ftp):\\/)?\\/?([^:\\/\\s]+)((\\/\\w+)*\\/)
               ([\\w\\-\\.]+[^#?\\s]+)(.*)?(#[\\w\\-]+)?\\b",
    "description": "Matching URLs like: http://www.test.com/dir
                    /filename.jpg?var1=foo#bar&var2=val2"
  },
  {
    "regex_id": "Geo-Coordinate",
    "pattern": "\\b([-+]?)([\\d]{1,2})(((.)(\\d+)(,)))(\\s*)(([-+]?)
               ([\\d]{1,3})((.)(\\d+))?)\\b",
    "description": "Matching Geo-Coordinates like: 51.498134, -0.201755"
  },
  {
    "regex_id": "CanadianSIN",
    "pattern": "\\b(\\d{3}[\\s]\\d{3}[\\s]+\\d{3})\\b|\\b(\\d{3}[-]\\d{3}[-]+\\d{3})\\b",
    "description": "Matching Canadian Social Insurance Number like: 123-456-789"
  },
  {
    "regex_id": "CreditCardExpirationDate",
    "pattern": "\\b\\d{2}\\/\\d{2}\\b",
    "description": "Matching Credit Card Expiration Date like 11/12"
  },
  {
    "regex_id": "CVV-Number",
    "pattern": "\\b[0-9]{3,4})\\b",
    "description": "Matching Credit Card Verification Value number like: 670, 0927"
  },
  {
    "regex_id": "Currency",
    "pattern": "\\b(\\d+(\\.\\d{2})?)\\b",
    "description": "Matching currency like: 123, 25.50"
  }
]
```


/policyengine/v1/regex/<regex_name>: GET

Retrieves a single regular expression detail from the IBM Data Cataloging system.

The following table shows the roles that can perform read operation on the endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓	✓	X	X

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer ${TOKEN}" https://$SDHOST/policyengine/v1/regex/$REGEXNAME -H "Content-type:application/json"
-X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example returns the single regular expression detail registered for which you have read permission.

1. Issue the following request in one line.

   ```
   curl -k -H "Authorization: Bearer ${TOKEN}" https://$SDHOST/policyengine/v1/regex/EmailID -H "Content-
   type:application/json" -X GET
   ```

2. The following response is returned.

   ```
   200(OK)

   {
           "regex_id": "EmailID",
           "pattern": "\\b[\\w\\.=-]+@[\\w\\.-]+\\.[\\w]{2,3}\\b",
           "description": "Matching Email IDs like : John.Smith@example.com"
   }
   ```

/policyengine/v1/regex: PUT

Updates the regular expressions in the IBM Data Cataloging system.

The table shows the roles that can perform update operation on the endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓	✓	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>'
https://$SDHOST/policyengine/v1/regex -d @regexp.json -H "Content-type:application/json" -X PUT
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

The following example shows how to update a regular expression.

Issue the following request on one line:

```
curl -k -H "Authorization: Bearer ${TOKEN}" https://$SDHOST/policyengine/v1/regex -d @regexp.json -H "Content-
type:application/json" -X PUT
```

Example: Request JSON payload

```
{
  "regex_id": "US-SSN-blank-delimited",
  "pattern": "\\b\\d{3}\\s\\d{2}\\s\\d{4}\\b",
  "description": "US SSN delimited by white space, not dashes."
}
```

The following response is returned:

```
200(OK)
Accepted
```

/policyengine/v1/regex: DELETE

Deletes the regular expression with value from the IBM Data Cataloging system.

The table shows the roles that can perform delete operation on the endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	X	X	X	X

Synopsis of the request URL

```
curl -k -s -o response.txt -H "Authorization: Bearer ${TOKEN}" https://${SDHOST}/policyengine/v1/regex/$REGEX_NAME -X DELETE
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

The following example shows how to delete a regular expression record.

Issue the following delete request:

```
curl -k -s -o response.txt -H "Authorization: Bearer ${TOKEN}" https://${SDHOST}/policyengine/v1/regex/US-SSN-blank-delimited -
X DELETE
```

The following response is returned:

```
204(No Content)
```

Note: When you perform DELETE operation, the resource gets deleted successfully but there is no response message in the output.

Adding fields with the AUTOTAG action

With the AUTOTAG action you can create policies that automatically add fields either to a specified set of documents or to all the documents in the system.

Two methods are available for adding new fields. One method is to add fields that contain initial values. The other method is to add fields that contain values that are extracted from existing fields of the same document. The following topics describe these two methods.

For more information about creating a policy, see the topic /policyengine/v1/policies -d '<data>': POST.

- **Adding fields that contain initial values**
  With the AUTOTAG action you can create a policy that automatically adds fields with initial values to a specified set of documents.
- **Adding fields that are extracted by rule from existing fields**
  Create a new field whose value is extracted from an existing field.

Adding fields that contain initial values

With the AUTOTAG action you can create a policy that automatically adds fields with initial values to a specified set of documents.

This method is useful when you want to add fields to a set of documents and you know beforehand the values that you want the fields to contain. For example, you might want to add a "billingaddress" field that contains the initial value "100 Corporation Street, Metropolis, USA" to every document that has a "username" field that contains the value "mjsmith". Or you might want to add an "ownerid" field that contains the initial value "admin02" to every document that lies in a certain directory path.

For more information about creating a policy, see the topic /policyengine/v1/policies -d '<data>': POST.

The following example shows how to create a policy that adds a project number and a department code to each document that contains a particular user name:

```
$ curl -k  -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/policyengine/v1/policies/autotagpol1
-d '{
"pol_filter":"user='research1'",
"action_id": "AUTOTAG",
"action_params": {
    "tags": {"ProjectNumber":"1404", "DepartmentCode":"H8AC"}
    }
```

```
"schedule":"NOW"
}'
-X POST -H "Content-Type: application/json"
```

The preceding example adds the fields "ProjectNumber":"1404" and "DepartmentCode":"H8AC" to every document in which the "user" field is set to 'research1'.

Adding fields that are extracted by rule from existing fields

Create a new field whose value is extracted from an existing field.

With the `AUTOTAG` action you can create a policy that adds a field to a document, where the new field contains a value that is extracted by rule from an existing field in the same document. Two types of rule are available. With the first type of rule, you can extract a value from an existing field by parsing tokens in the field value that are separated by a specified delimiter. With the second type of rule, you can extract a value from an existing field by searching the field value with a regular expression.
Note: The new field can contain a maximum of 256 characters. If the extracted value contains more than 256 characters, it is trimmed to 256 characters before it is assigned to the new field.
In the following two examples, the documents contain a `"filepathActiveUsers"` field in which the parts of the path are separated by forward slashes (/) and in which the second part of the path contains the project name. For example:

```
"filepathActiveUsers":"'/Lab1/ProjectHA48/users/active'"
```

You want to add a field `"projectname"` that contains the project name, and you want to add the field to all documents that contain the field `"user":"'research1'"`.

1. The first example shows how to extract the project name by parsing the path into tokens that are separated by a forward slash. Follow these steps:
   a. Create a file autotag_rule_pol_split.dat that contains the parameters for the new policy. The following code block shows the contents of the file:

```
{
    "action_id": "AUTOTAG",
    "action_params":
    {
        "rule" : {
            "name": "setFromExistingField",
            "action": "split",
            "existingField": "filepathActiveUsers",
            "delimiter": "/",
            "fieldNo": 2,
            "newField": "projectname"
        }
    },
    "schedule": "NOW",
    "pol_filter": "user='research1'
}
```

   b. Create the policy and apply it:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/policyengine/v1/policies/autotagpol2
-d @autotag_rule_pol_split.dat
-X POST
-H "Content-Type: application/json"
```

   The policy extracts the second token, the project name, from the path and assigns the value to the new field `"projectname"`.

2. The second example shows how to extract the project name from the path with a regular expression. Follow these steps:
   a. Create a file autotag_rule_pol_regex.dat that contains the parameters for the new policy. The following code block shows the contents of the file:

```
{
    "action_id": "AUTOTAG",
    "action_params": {
        "rule" : {
            "name": "setFromExistingField",
            "action": "regex",
            "existingField": "filepathActiveUsers",
            "regexPattern": "[^/]+",
            "matchNo": 3,
            "newField": "projectname"
        }
    },
    "schedule": "NOW",
    "pol_filter": "user='research1'"
}
```

   b. Create the policy and apply it:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/policyengine/v1/policies/autotagpol2
-d @autotag_rule_pol_regex.dat
-X POST
-H "Content-Type: application/json"
```

   The policy extracts the third match to the regex pattern, the project name, from the path and assigns the value to the new field `"projectname"`.

Endpoints for working with connection management

With the connection management API service you can create, update, delete, and get information about data source connection entries in the connection table.

A data source connection contains the following attributes:

name
: Indicates the name of the connection.

type
: Indicates the type of the connection.

cluster
: Indicates the cluster in which the data source is located.

datasource
: Indicates the file system or vault in which the data is stored.

site
: Indicates the location in which the cluster is maintained.

Note: To determine cluster and data source values for a connection in an IBM Spectrum Scale cluster, follow these steps:

1. Open a console on a node in the IBM Spectrum Scale cluster.
2. Run the following command to get information about the cluster:

   ```
   /usr/lpp/mmfs/bin/mmlscluster
   ```

   The command output displays information about the cluster similar to the following example:

   ```
   GPFS cluster information
   ========================
     GPFS cluster name:         modevvm19.metro.labs.cpr.com
     GPFS cluster id:           7146749509622277333
     GPFS UID domain:           modevvm19.metro.labs.cpr.com
     Remote shell command:      /usr/bin/ssh
     Remote file copy command:  /usr/bin/scp
     Repository type:           CCR

    Node  Daemon node name               IP address   Admin node name               Designation
   ----------------------------------------------------------------------------------------------
      1   modevvm19.metro.labs.cpr.com  9.11.201.74  modevvm19.metro.labs.cpr.com quorum-manager
   ```

3. Make a note of the IBM Spectrum Scale cluster name, which in the preceding example is `modevvm19.metro.labs.cpr.com`. Use this value as the `cluster` attribute for the connection.
4. Run the following command to get information about all the mounted file systems in the cluster:

   ```
   /usr/lpp/mmfs/bin/mmlsmount all
   ```

   The command output displays information about mounted file systems as in the following example:

   ```
   File system gpfs0 is mounted on 1 nodes.
   ```

5. Make a note of the name of the file system that you want to create a connection for. For example, in the preceding example only one file system is mounted and its name is `gpfs0`. Use this value as the `datasource` attribute for the connection.

- **/connmgr/v1/connections: GET and /connmgr/v1/connections/<connection_name>: GET**
  Lists the attributes of one or more connections.
- **/connmgr/v1/scan/<connection>/partitions: GET**
  Gets the list of fileset (IBM Storage Scale) or Shares (SMB/CIFS) on the target system.
- **/connmgr/v1/connections -d '<data>': POST**
  Creates a connection.
- **/connmanager/v1/scan/<connection>/partial: POST**
  Initiates a partial scan of the datasource connection.
- **/connmgr/v1/connections/<connection_name> -d '<data>': PUT**
  Updates a connection.
- **/connmgr/v1/connections/<connection_name>: DELETE**
  Deletes a connection.
- **/connmgr/v1/scan/<connection_name>: POST**
  Starts the scan of a connection.
- **/connmgr/v1/scan/<connection_name>: GET**
  Gets the current scan status of a single connection.
- **/connmgr/v1/scan: GET**
  Gets the scan status of all running scans.
- **/connmgr/v1/scan/<connection_name>: PUT**
  Stops a scan.

/connmgr/v1/connections: GET and /connmgr/v1/connections/<connection_name>: GET

Lists the attributes of one or more connections.

These two endpoints list the attributes either of a specified connection or of all the connections in the connection table. The following table shows which roles can access these two REST API endpoints:

Table 1. Access by role

Endpoints	Data admin	Data user	Collection Admin	Admin	Service user
/connmgr/v1/connections: GET	✓	X	✓	X	X
/connmgr/v1/connections/<connection_name>: GET	✓	X	✓	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/connmgr/v1/ connections -X GET
```

or

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/connmgr/v1/ connections/<connection_name> -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON
- text/plain

Examples

1. The following example returns information about all the connections in the connection table:
   a. Run the following request on one line:

   ```
   curl -k -H 'Authorization: Bearer <token>'
   https://<data_cataloging_host>/connmgr/v1/connections -X GET -H 'Accept: application/json'
   ```

   The following response is returned. The response lists the information for all the connections in the connections table. For example, this table contains two connections, `con01` and `con02`:

   ```
   [
      {
         "name": "con01",
         "type": "general",
         "cluster": "modevvm19.metro.labs.cpr.com",
         "datasource": "gpfs0",
         "site":"datasite03"
      }
      {
         "name": "con02",
         "type": "general",
         "cluster": "modevvm19.metro.labs.cpr.com",
         "datasource": "gpfs1",
         "site":"datasite03"
      }
   ]
   ```

2. The following example returns information about connection `con02`:
   a. Run the following request on one line:

   ```
   curl -k -H 'Authorization: Bearer <token>'
   https://<data_cataloging_host>/connmgr/v1/connections/con02 -X GET -H 'Accept: application/json'
   ```

   The following response is returned:

   ```
   [
      {
         "name": "con02",
         "type": "general",
         "cluster": "modevvm19.metro.labs.cpr.com",
         "datasource": "gpfs1",
         "site":"datasite03"
      }
   ]
   ```

/connmgr/v1/scan/<connection>/partitions: GET

Gets the list of fileset (IBM Storage Scale) or Shares (SMB/CIFS) on the target system.

This API endpoint fetches the list of filesets (IBM Storage Scale) or shares (SMB/CIFS) on the target system. Only an IBM Data Cataloging user with the data admin role can get the list of partitions.

The following table displays the roles that can access the REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	X	[1]√	X	X
[1] The Collection Admin user, can list the partitions for connections assigned to the collections which are assigned to them.				

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/connmgr/v1/scan/<connection_name>/partitions -X GET
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

1. The following example returns partition (fileset) information from an IBM Storage Scale connection:
   - Run the following request on one line:

     ```
     curl -k -H 'Authorization: Bearer <token>'
     https://< data_cataloging_host>/connmgr/v1/scan/ss1 -X GET -H 'Accept: application/json'
     ```

   ```
   ["root", "fileset1", "fileset2"]
   ```

2. The following example returns partition (share) information from an SMB/CIFS connection:
   Run the following request on one line:

   ```
   curl -k -H 'Authorization: Bearer <token>'
   https://<data_cataloging_host>/connmgr/v1/scan/smb1 -X GET -H 'Accept: application/json'
   ```

   ```
   ["root", "dir1"]
   ```

/connmgr/v1/connections -d '<data>': POST

Creates a connection.

The /connmgr/v1/connections -d *'<data>'*: POST endpoint creates a new connection entry in the connections table. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/connmgr/v1/
connections/ -d '<data>' -X POST -H "Content-Type: application/json"
```

Supported request types, input fields, and response formats

Supported request types:

- POST

Supported input fields:
JSON input is expected. Use a **Content type: application/json** header in the HTTP protocol request.

name
: The unique name of this connection.

cluster
: The cluster ID of the IBM Storage Scale cluster or IBM Cloud® Object Storage system UUID.

platform
: ["Spectrum Scale" | "IBM® COS" | "NFS" | "S3" | "IBM Storage Protect"] | "SMB/CIFS"

datasource
: Name of individual filesystem or vault. (For example, `scratch`)

site
: Physical location of data from this connection. This field is optional.

host
: The hostname or IP address of the interface node for scanning. This field is not mandatory for creating a connection, but is required for automated scan support.

mount_point
: The network path to be scanned. This field is only used for NFS and SMB/CIFS connections. For any other connection types, it is an optional field. For NFS, the 'mount_point' is the export path, for example, /export1/dir1. For SMB/CIFS, the 'mount_point' is the network share path, for example, \\SMBSERVER\share1\.

user
: The username to connect to the source data system with. This field is not mandatory for creating a connection but is required for automated scan support.

password
: The password on the source data system belonging to the user ID specified earlier. This field is not mandatory for creating a connection, but is required for automated scan support for connections other than NFS.

additional_info
: Connection-type specific information. This field is not mandatory for creating a connection, but is required for automated scan support. This information varies by platform as follows:

  IBM Cloud Object Storage (IBM COS)

  accesser_address
  : The hostname or IP address of the IBM COS accessor system.

accesser_access_key
: the access key for the corresponding accessor address specified by the `accesser_address`.

accesser_secret_key
: The accessor's secret key, corresponding to the IBM COS `accesser_address`.

manager_username
: User name of the manager for the IBM COS system.

manager_password
: Password of the IBM COS manager user name specified earlier.

SMB/CIFS

auth_type
: Denotes SMB authentication type parameter.

IBM Storage Scale

working_dir
: Directory on the IBM Storage Scale system to be used for the IBM Data Cataloging files used during scans of the IBM Storage Scale file system. This directory must exist before the connection is established.

nodes
: Specify a list of one or more nodes in the IBM Storage Scale cluster or the keyword `all` to enable the IBM Data Cataloging scans to execute across all nodes. The node names must match the names used by the IBM Storage Scale cluster configuration. For example, the nodes shown by the **mmlscluster** command executed on a node.

scan_dir
: The directory path within the IBM Storage Scale file system that is to be scanned. For example, `/scale/zoo`.

NFS

NFSv4
: Denotes NFS protocol.

S3

access_key
: Access key for the S3 data connection.

secret_key
: The secret key corresponding to this S3 data connection.

IBM Storage Protect

port
: The Open Database Connector (ODBC) port for the IBM Storage Protect server. This field is set to `51500` by default.

Supported response formats:

- JSON

Examples

1. The following example creates a connection that is named con01:
   a. Issue the following request on one line:

   ```
   curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/connmgr/v1/
   connections -d '{"name":"con01","platform":"Spectrum Scale", "cluster":
   "modevvm19.metro.labs.cpr.com", "datasource": "gpfs1", "site":"datasite03"}' -X POST
   -H "Content-Type: application/json"
   ```

   b. The following response is returned:

   ```
   Connection 'con01' added
   ```

   To verify that the connection is created, issue a GET request to list the information about the specified connection or about all the connections in the connection table. For more information, see /connmgr/v1/connections: GET and /connmgr/v1/connections/<connection_name>: GET.
2. The following examples demonstrates to create a connection using SMB parameter `auth_type` and NFSv4 protocol:
   - POST request using SMB parameter `auth_type`

     ```
     curl -s -k -H "Authorization: Bearer <token> -H "Content-Type: application/json" -X POST
     http://<data_cataloging_host>/connmgr/v1/
     connections -d'{"name":"perffs","platform":"Spectrum Scale","cluster":"gkscale4app-nd-
     1.fyre.ibm.com","datasource":"perffs","site":"svl","host":"host_ip","user":"root","password":"P@ssw0rd","additional_i
     nfo":"{"working_dir":"/gpfs/perffs","nodes":"all","scan_dir":"/gpfs/perffs","auth_type":"password"}"}'
     ```

   - POST request using NFS protocol `NFSv4`

     ```
     curl -s -k -H "Authorization: Bearer <token> -H "Content-Type: application/json" -X POST
     <data_cataloging_host>/connmgr/v1/
     connections -
     d'{"name":"nfs4test","platform":"NFS","cluster":"cluster1","datasource":"nfs4test","site":"Tucson","host":"
     <host_ip>","mount_point":"/ifs/sdiscover/Data/nfs4test","protocol":"nfs"}'
     ```

/connmanager/v1/scan/<connection>/partial: POST

Initiates a partial scan of the datasource connection.

This API endpoint initiates a partial scan of the datasource connection by specifying a list of one or more partitions to be scanned.

The table shows the roles that can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	[1]✓	X	X
[1]Collection Admin users can initiate a partial scan of connections for collections to which they have been assigned.				

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/connmanager/v1/scan/connection/partial -X POST -H 'Content-type: application/json'
```

Supported request types and response formats

Supported request types:

- POST

Supported input fields:

- JSON input in the following form:

  ```
  {"partitions": ["partition1", "partition2", "etc."]}
  ```

- JSON input in the following form for NFS partial scanning:

  ```
  {"partitions": [{"dir": "/exports/example/a", "recursive": true}, {"dir": "/exports/example/b", "recursive": false}]}
  ```

  where each subdirectory to scan is an item of the array with the following structure:

  ```
  {
      "dir": "/exports/example/a", // Directory path that starts with the mount point of the NFS connection
      "recursive": false // true if directories inside of the provided path should be scanned recursively
  }
  ```

Supported response formats:

- JSON
  Returns the status of whether or not the scan was started successfully. Error messages are displayed if JSON is malformed or if a specified partition is not found on the filesystem.

Examples

Issue the following request in one line:

```
curl -k -H 'Content-Type: application/json' -X POST
https://<data_cataloging_host>/connmgr/v1/scan/smb1/partial -H "Accept:application/json" -H "Content-type: application/json"
-X POST -d'{"partitions": ["dir1"]}
```

Issue the following NFS partial scan request:

```
curl -k -H 'Content-Type: application/json' -X POST
https://<data_cataloging_host>/connmgr/v1/scan/nfs-1/partial -H "Accept:application/json" -H "Content-type: application/json"
-X POST -d'{"partitions":[{"dir":"/exports/data","recursive":true}]}'
```

The following response is returned:

```
{"status": "Success"}
```

/connmgr/v1/connections/<connection_name> -d '<data>': PUT

Updates a connection.

The /connmgr/v1/connections/*<connection_name>* -d *'<data>'*: PUT endpoint updates an existing connection with the attribute values that are specified in the request data. Attributes that are not updated keep the same values that they had before the update. The following attributes can be updated: **`name`**, **`type`**, **`cluster`**, **`datasource`**, and **`site`**. For more information about the attributes in the request data, see Endpoints for working with connection management. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/connmgr/v1/ connections/<connection_name> -d '<data>'
-X PUT -H "Content-Type: application/json"
```

Supported request types, input fields, and response formats

Supported request types:

- PUT

Supported input fields:
JSON input is expected. Use a **Content type: application/json** header in the HTTP protocol request.

name
: The unique name of this connection.

cluster
: The cluster ID of the IBM Storage Scale cluster or IBM Cloud® Object Storage system UUID.

platform
: ["Spectrum Scale" | "IBM® COS" | "NFS" | "S3" | "IBM Storage Protect"].

datasource
: Name of individual file system or vault. For example, `scratch`.

site
: Physical location of data from this connection. This field is optional.

host
: The hostname or IP address of the interface node for scanning. This field is not mandatory for creating a connection, but is required for automated scan support.

user
: The username to connect to the source data system with. This field is not mandatory for creating a connection, but is required for automated scan support.

password
: The password on the source data system belonging to the user ID specified in the previous field. This field is not mandatory for creating a connection, but is required for automated scan support.

additional_info
: Connection-type specific information. This field is not mandatory for creating a connection, but is required for automated scan support.

Supported response formats:

- JSON

Examples

1. The following example updates connection `con01` to `con02`:
   a. Issue the following request on one line:

   ```
   curl -k -H 'Authorization: Bearer <token>'https://<data_cataloging_host>/connmgr/v1/
   connections/con01 -d'{"name":"con02","platform":"Spectrum Scale","cluster":
   "modevvm19.metro.labs.cpr.com","datasource":"gpfs1", "site":"datasite03"}' -X PUT
   -H "Content-Type:application/json"
   ```

   b. The following response is returned:

   ```
   Connection 'con02' updated
   ```

   To verify that the policy is updated, issue a GET request to list the information about the specified connection or about all the connections in the system. For more information, see /connmgr/v1/connections: GET and /connmgr/v1/connections/<connection_name>: GET.

/connmgr/v1/connections/<connection_name>: DELETE

Deletes a connection.

The /connmgr/v1/connections/*<connection_name>*: DELETE endpoint deletes the specified connection from the connection table. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/connmgr/v1/ connections/<connection_name> -X DELETE
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

1. The following example deletes a connection named con02:
   a. Issue the following request on one line:

   ```
   curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/connmgr/v1/connections/con02 -X DEL
   ```

   b. The following response is returned:

   ```
   Connection 'con02' deleted
   ```

   To verify that the connection is deleted, issue a GET request to list the information about the specified connection or about all the connections in the connection table. For more information, see /connmgr/v1/connections: GET and /connmgr/v1/connections/<connection_name>: GET.

/connmgr/v1/scan/<connection_name>: POST

Starts the scan of a connection.

The /connmgr/v1/scan/*<connection_name>*: POST endpoint starts the scan of a connection. Only an IBM Data Cataloging user with data admin role or collection admin role can start a scan. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	X	√[1]	X	X
[1] Collection Admin user can start, stop, and get the status of the scans that are applied to the collections to which they have the Collection Admin role assigned.				

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer <token>" -X POST https://<data_cataloging_host>/connmgr/v1/scan/<connection_name>
```

Supported request types, input fields, and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

Follow these steps to start a scan:

1. Obtain an auth token by using the credentials of the data admin user as shown:

   ```
   curl -k https://<data_cataloging_host>/auth/v1/token -u "<user_name>:<password>
   ```

   For a valid user, the auth token is returned in the **X-Auth-Token** response header.

2. Start the scan of a connection with name **sdconnection** using the following request:

   ```
   curl -k -H "Authorization: Bearer <token>" -X POST https://<data_cataloging_host>/connmgr/v1/scan/sdconnection
   ```

Response

The following table displays the common response statuses:

Table 2. Common response statuses

Status code	Message	Description
400	Must specify a connection name for the scan endpoint.	Connection name in the URL is not specified.
	Scan already currently in-progress for *<connection_name>*.	Scan already in progress.
404	Could not locate connection document for *<connection_name>*. Aborting scan.	Connection with the given name does not exist.
403	User does not have permission to view connection.	User does not have access to the collection to which the connection belongs.
200	-	Scan successfully started.

Note: A user with a collection admin role can only start a scan for a connection if an associated collection is set, and if it is a collection the collection admin user administers.
A successful scan displays the following response:

```
"Status": "Success"
```

The following displays an error response example:

```
"status": "Connection name in URL not specified."
```

/connmgr/v1/scan/<connection_name>: GET

Gets the current scan status of a single connection.

The /connmgr/v1/scan/*<connection_name>*: GET endpoint gets the current scan status of a single connection. Only an IBM Data Cataloging user with data admin role or collection admin role can get the status of a scan. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	X	√[1]	X	X
[1] Collection Admin user can start, stop, and get the status of the scans that are applied to the collections to which they have the Collection Admin role assigned.				

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer <token>" -X GET https://<data_cataloging_host>/connmgr/v1/scan/<connection_name>
```

Supported request types, input fields, and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

Follow these steps to get the scan status of a connection:

1. Obtain an auth token by using the credentials of the data admin user as shown:

   ```
   curl -k https://<data_cataloging_host>/auth/v1/token -u "<user_name>:<password>
   ```

   For a valid user, the auth token is returned in the **X-Auth-Token** response header.

2. Get the scan status of a connection with name **sdconnection** using the following request:

   ```
   curl -k -H "Authorization: Bearer <token>" -X GET https://<data_cataloging_host>/connmgr/v1/scan/sdconnection
   ```

Response

The following table displays the common response statuses:

Table 2. Common response statuses

Status code	Message	Description
200	Scan status JSON.	Scan status successfully retrieved.

Note: A user with a collection admin role can only get the scan status for a connection if an associated collection is set, and if it is a collection the collection admin user administers. Therefore, in the response, a collection admin user is only presented with the status of connections that belong to the connections the collection admin user administers.
A successful scan displays the following response:

```
[
{
"Name":"sdconnection",
"status":"Running",
"message":"Crawling NFS mount",
"Phase":2,
"Total_phases":3
},
{
"Name":"ss",
"status":"Running",
"message":"Preparing to scan connection",
"Phase":1,
"total_phases":4
}
]
```

/connmgr/v1/scan: GET

Gets the scan status of all running scans.

The /connmgr/v1/scan endpoint gets the current scan status of all connections. Only an IBM Data Cataloging user with data admin role or collection admin role can get the status of all running scans. The following table shows which roles can access /connmgr/v1/scan endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	✓[1]	X	X
[1] Collection Admin user can start, stop, and get the status of the scans that are applied to the collections to which they have the Collection Admin role assigned.				

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer <token>" -X GET https://<data_cataloging_host>/connmgr/v1/scan
```

Supported request types, input fields, and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

Follow these steps to get the scan status of all the running scans:

1. Obtain an auth token by using the credentials of the Data Admin user as shown:

   ```
   curl -k https://<data_cataloging_host>/auth/v1/token -u "<user_name>:<password>
   ```

   For a valid user, the authentication token is returned in the **X-Auth-Token** response header.

2. Get the scan status of all the running scans by using the following request:

   ```
   curl -k -H "Authorization: Bearer <token>" -X GET https://<data_cataloging_host>/connmgr/v1/scan
   ```

Response

The following table displays the common response statuses:

Table 2. Common response statuses

Status code	Message	Description
404	No scan is in progress for *<connection_name>*.	No scan is in progress for the specified connection.
403	User does not have permission to get scan status for connection *<connection_name>*.	User cannot access the collection to which the connection belongs.
200	Scan status JSON.	Scan status successfully retrieved.

Note: A user with a collection admin role can get only the current scan status for a connection if an associated collection is set, and if it is a collection the Collection Admin user administers.
A successful scan displays the following response:

```
{
"Name":"sdconnection",
"status":"Running",
"message":"Crawling NFS mount",
"Phase":2,
"total_phases":3
}
```

The following displays an error response example:

```
{
        "status": "Connection name in URL not specified."
}
```

/connmgr/v1/scan/<connection_name>: PUT

Stops a scan.

The /connmgr/v1/scan/*<connection_name>*: PUT endpoint stops a running scan. Only an IBM Data Cataloging user with data admin role or collection admin role can stop a scan. The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	✓[1]	X	X
[1] Collection Admin user can start, stop, and get the status of the scans that are applied to the collections to which they have the Collection Admin role assigned.				

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer <token>" -X PUT https://<data_cataloging_host>/connmgr/v1/scan/<connection_name>/stop
```

Supported request types, input fields, and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

Follow these steps to stop a scan:

1. Obtain an auth token by using the credentials of the data admin user as shown:

   ```
   curl -k https://<data_cataloging_host>/auth/v1/token -u "<user_name>:<password>
   ```

   For a valid user, the auth token is returned in the **X-Auth-Token** response header.

2. Stop the scan using the following request:

   ```
   curl -k -H "Authorization: Bearer <token>" -X PUT https://<data_cataloging_host>/connmgr/v1/scan/sdconnection/stop
   ```

Response

The following table displays the common response statuses:

Table 2. Common response statuses

Status code	Message	Description
400	Must specify a connection name for the scan endpoint.	Connection name in the URL is not specified.
	Must specify an action for the scan endpoint. Available action endpoints are: **/pause**, **/resume**, and **/stop**.	The stop action is not specified in the URL.
	No scan is currently in progress for *<connection_name>*.	No scans currently in progress for the given connection.
	Unsupported action **<action specified in url>**.	Unsupported action specified.
404	Could not locate connection document for *<connection_name>*.	Connection with the given name does not exist.
403	User does not have permission to control connection.	User does not have access to the collection to which the connection belongs.
200	-	Scan successfully stopped.

Note: A user with a collection admin role can only stop a scan for a connection if an associated collection is set, and if it is a collection the collection admin user administers.
A successful scan displays the following response:

```
"Status": "Stop of sdconnection scan submitted."
```

The following displays an error response example:

```
"Status": "No scan is currently in progress for sdconnection."
```

Application management by using APIs

The application management APIs provide options to get the details of the available applications, install or upgrade an application, and delete an application. You must have the Admin role to access the application management endpoints.

You can change the self-signed certificate that IBM Data Cataloging uses. For more information, see IBM Data Cataloging REST APIs.For more information, see the topic *IBM Spectrum® Discover REST APIs* in the *Data Cataloging: Administration Guide*.

- **/policyengine/v1/applications: POST**
  The /policyengine/v1/applications registers an action agent. Only an IBM Data Cataloging user with *data admin* role can register the application with IBM Data Cataloging.
- **/policyengine/v1/applications/<application name>: GET**
  Gets the details of a specific application that is configured in the system.
- **/policyengine/v1/tlscert: GET**
  Gets a CA-certified TLS certificate.
- **/policyengine/v1/action_ids: GET**
  Retrieves and displays a list of actions IDs of all the registered applications.
- **/policyengine/v1/applications/<deployment_name>/schema?action_id=<action_id>: GET**
  Gets the application schema for a specified application.
- **/policyengine/v1/application_names?action_id=<action_id>: GET**
  Gets a list of applications that support a specified action ID.
- **/policyengine/v1/applications: DELETE**
  Deletes an application.

/policyengine/v1/applications: POST

The /policyengine/v1/applications registers an action agent. Only an IBM Data Cataloging user with *data admin* role can register the application with IBM Data Cataloging.

The following table shows which roles can access /policyengine/v1/applications endpoint:

Table 1. Access by role

Data admin	Collection Admin	Data user	Admin	Service user
√	X	X	X	X

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer <token>" -H "Content-Type: application/json" -X POST
https://<data_cataloging_host>/policyengine/v1/applications
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

Registering an application involves the following steps.

1. Obtain an auth token by using credentials of the data admin user as shown in the following example:

   ```
   curl -k https://<data_cataloging_host>/auth/v1/token -u "<user_name>:<password>
   ```

   For a valid user, the auth token is returned in the **"X-Auth-Token"** response header.

2. Create a .json file with application details:

   ```
   cat agentregmsg.json

   {
   "action_agent": "extractapplication",
   "action_id": "DEEPINSPECT",
   "action_params": ["extract_tags"]
   }
   ```

   Note: Only **DEEPINSPECT** is supported as a valid **action_id** and must always be in upper case. The **action_agent** parameter must be a maximum of 64 characters long and can contain alpha-numeric characters and '.', '_' or '-' characters. However, the action parameters that are converted to string can be a maximum of 256 characters long.
3. Submit the following request:

   ```
   curl -k https://<data_cataloging_host>/policyengine/v1/applications -H "Content-Type: application/json" -X POST -d
   @agentregmsg.json -H "Authorization: Bearer <token>"
   ```

Response:

```
Content-Type: application/json
    {
      "broker_ip": 9.11.200.114,
      "broker_port": 9093,
      "work_q": "extractapplication_work",
      "completion_q": "extractapplication_compl"
    }
```

Multiple application instances

If you want to run multiple copies of an application, then only the first application registration attempt receives a success response. Any subsequent applications receive a **409 Conflict** response. In such cases, use the /policyengine/v1/agents/<agent name> endpoint to find the required Kafka information to proceed.

Example GET response:

```
{
"broker_ip": "localhost",
"work_q": "extractapplication_work",
"auth": "extractapplication_user:extractapplication_password",
"params": "["tags"]",
"agent": "extractagent",
"broker_port": "9093",
"completion_q": "extractapplication_compl",
"action_id": "\"DEEPINSPECT\""
}
```

- **Example: Create a DeepInspect policy using the application**
  Use the POST /policyengine/v1/policies/extractpol endpoint to create a policy that makes use of the registered application.

Example: Create a DeepInspect policy using the application

Use the POST /policyengine/v1/policies/extractpol endpoint to create a policy that makes use of the registered application.

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer <token>" -H "Content-Type: application/json" -X POST
https://<data_cataloging_host>:443/policyengine/v1/policies/extractpol
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

Registering an application involves the following steps.

1. Define the policy details in a JSON file as shown in the following example:

```
cat testpol
{
        "action_params": {
                "extract_tags": ["vin", "sensor"],
                "agent": "extractapplication"
        },
        "pol_id": "extractpol",
        "schedule": "NOW",
        "pol_filter": "size>23",
        "action_id": "DEEPINSPECT"
}
```

2. Submit the following request:

```
curl -k-H "Authorization: Bearer <token>"https://<data_cataloging_host>/policyengine/v1/policies/extractpol -X POST -d
@testpol -H "Content-Type: application/json"
```

As the policy is created, as the policy schedule is 'NOW'. The job request messages are immediately pushed to the Kafka work topic corresponding to the registered application as shown in the following example:

The following example provides a JSON message request.

```
{
    "mo_ver": "1.0",
    "action_id": "deepinspect",
    "action_params": {
        "agent": "extractapplication",
          "tags": {"extract_tags": ["vin", "sensor"]}
    },
    "agent": " extractapplication",
    "policy_id": " extractpol",
    "docs": [
        {"path": "/fs1/path1/file1.txt", "fkey": "spectrumscale.cluster.example"},
      {"path": "/fs1/path1/file2.txt", "fkey": "spectrumscale.cluster.example"},
            ......
          {"origpath": "/fs1/path1/file3.txt", "fkey": "spectrumscale.cluster.example"}
    ]
}
```

The following table lists the job request message format.

Table 1. Job request message format

Field	Value Type	Description
mo_ver	Float	The message version
policy_id	String	The name of the policy ID that requested the job.
action_id	String	The name of the action ID
agent	String	The name of the application
action_params	Object	JSON object of custom application parameters
docs	Array	The array of JSON objects, each containing information about documents (files or objects) to be inspected.

The following example provides a JSON message response:

```
{
    "mo_ver": "1.0",
    "policy_id": " extractpol",
    "docs": [
        {"status": "success", "tags": {"vin": "vin-value", "sensor": "sensor-value"}, "path": "/fs1/path1/file1.txt", "fkey":
"spectrumscale.cluster.example "},
        {"status": "success", "tags": {"vin": "vin-value", "sensor": "sensor-value"}, "path": "/fs1/path1/file1.txt", "fkey":
"spectrumscale.cluster.example "},
        {"status": "failed", "tags": {}, "path": "/fs1/path1/file1.txt", "fkey": "spectrumscale.cluster.example "}

    ]
}
```

The following table lists the job response message format.

Table 2. Job response message format

Field	Value Type	Description
mo_ver	Float	The message version
policy_id	String	The name of the policy ID that requested the job.
docs	Array	The array of JSON objects, each containing information about documents (files/objects) to be inspected.

/policyengine/v1/applications/<application name>: GET

Gets the details of a specific application that is configured in the system.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	CollectionAdmin	Data user	Admin	Service user
✓	X	✓	X	X

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer <token>" -H "Content-Type: application/json"
https://<data_cataloging_host>/policyengine/v1/application/<application _name>
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of a specific application.

Request:

```
curl -k -H "Authorization: Bearer <token>" -H "Content-Type: application/json"
"https://<data_cataloging_host>/policyengine/v1/application/extractapplication"
```

Response:

```
{
"broker_ip": "localhost",
"work_q": "
extractagent_work",
"auth": "extractapplication_user:extractapplication_password",
"params": "["tags"]",
"agent": "extractagent",
"broker_port": "9093",
"completion_q": "extractapplication_compl",
"action_id": "deepinspect"
}
```

/policyengine/v1/tlscert: GET

Gets a CA-certified TLS certificate.

The IBM Data Cataloging uses TLS protocol for encrypting the communication in-flight between the IBM Data Cataloging nodes (the Kafka topics) and the applications. It uses TLS client certificates to securely authenticate the applications. The certificates that are provided by the IBM Data Cataloging admin upon registration are used by applications to authenticate to the Kafka brokers in IBM Data Cataloging.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	CollectionAdmin	Data user	Admin	Service user
✓	X	✓	X	X

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer <token>" https://<data_cataloging_host>/policyengine/v1/ tlscert
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get a CA-certified TLS certificate.

Request:

```
curl -k -H "Authorization: Bearer <token>" https://<data_cataloging_host>/policyengine/v1/ tlscert
```

Response:

```
-----BEGIN CERTIFICATE-----
Samplej1CAmoCCQDoQCvwi/bLdzANBgkqhkiG9w0BQsFADCBmDELMAkGA1UEBhMCR0IxDjAMBgNV
BAgMBUhBTlRTMRAwDgYDVQQHDAdIdXJzbGV5MQwwCgYDVQQKDANJQk0xGTAXBgNVBAsMEHNwZWN0
cnVtZGlzY292ZXIxGTAXBgNVBAMMEHNwZWN0cnVtZGlzY292ZXIxIzAhBgkqhkiG9w0BCQEWFGls
YXdyZW5jZUB1ay5pYm0uY29tMB4XDTE4MTAxNTIyNTEzOVoXDTE5MTAxNTIyNTEzOVowbTELMAkG
A1UEBhMCR0IxDjAMBgNVBAgTBUhhbnRzMRAwDgYDVQQHEwdIdXJzbGV5MQwwCgYDVQQKEwNJQk0x
CzAJBgNVBAsTAklEMSEw4wYDVQQDExh0aG9yLnR1Yy5zdGdsYWJzLmlibS5jb20wggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQCJFq8t8rk2fZf86TQWTE6R0VhFxmn9daqWyTQGz56zqGnX
UxyF1wnIZQp7EGS3rBgHemV48X7xgxbshNVflcr0jANvgElB66JOaESJUz1m/s3B728qpYHO4Wom
2ii17hF3VdQhAvd72hd2kpJ3XVtp95yktLJ40Cr6x/4Kgsm8iKIiYYN3LbqqtfNB5CyKV9qNzTGb
ERPeSujS4HjUbaXQ+xeex6PhT6MDHsuSCnT6p6W+WoAw1C+W8eKpDIRiQFBPxzFM1buvJutQ+H8u
5ipXKgi6ki5qYeKr5pdZKI/tHgAhZnuQOAbnGjumqRr1Ddx/0y4Ay5ZPcyRYeZJaUtJXAgMBAAEw
DQYJKoZIhvcNAQELBQADggEBAIo8Su+9I3wnbypy+c9Gee9p0zIv5ksJTC9KmUziVF4dr8J8t5Mx
cA42yB2Q8hL5OyGgIyYm3Kq+tbaq1jUgqdcOzKKxMIKnS5ut6uifoL3oEq4Gx6Qsjo+RLduEAqZC
LrCVM5Nm8i9mY5g4NIK/o3G2Mud5MKeHc/XnMw4ZNzqfsEIkiG1DAgLHVJVokdo4gtoB7pN9bdtA
Qdm7nH/N5vJL7HNTtupcyHQqTIot+WCJ1YqiKqZVnMqqjFm4zMAbzrNzMqIFdSNkJn9IU+6LJn7v
CrOu6Whgkh+37Plc2mnViWRmD2+gsHcPOUhhsJMltB+RghqEjStF4TW4v29E36E=
-----END CERTIFICATE-----
Bag Attributes
    friendlyName: kafka_client
    localKeyID: 54 69 6D 65 20 31 35 33 39 36 34 33 39 31 31 37 39 31
    1.3.18.0.2.28.24: IBM_SDK_JAVA_8_PKCS12
Key Attributes: <No Attributes>
-----BEGIN PRIVATE KEY-----
SampleIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCJFq8t8rk2fZf8
6TQWTE6R0VhFxmn9daqWyTQGz56zqGnXUxyF1wnIZQp7EGS3rBgHemV48X7xgxbs
hNVf1cr0jANvgElB66JOaESJUz1m/s3B728qpYHO4Wom2ii17hF3VdQhAvd72hd2
kpJ3XVtp95yktLJ40Cr6x/4Kgsm8iKIiYYN3LbqqtfNB5CyKV9qNzTGbERPeSujS
4HjUbaXQ+xeex6PhT6MDHsuSCnT6p6W+WoAw1C+W8eKpDIRiQFBPxzFM1buvJutQ
+H8u5ipXKgi6ki5qYeKr5pdZKI/tHgAhZnuQOAbnGjumqRr1Ddx/0y4Ay5ZPcyRY
eZJaUtJXAgMBAAECggEAGXLSlk8GvuzVuCd6eeE0fLFGF73bkyeywUccl18XcPaE
n2pYUyg2MrM/HNVQPSB/XXQFPkxmamMRcFrAWYgQoxES2jIQZ2JcwP1uK6f+MA6B
F81g76PF0yfZUtkQ0aUeLO8L8Azi8GTuEh7nfa0g+Sz0MUcyEmdicyeJK15YIN1e
3Z0bl6kment4xvcLjPa2iFa1UJZ8gMNT6aS+NBaexjNlDW+NlftplRH6z0iqUObd
MeNvi7kiqOpre/QiPUzJSmmPsDSwCCtob9oAHa9JrfLH19VnBQGqee4T/KPQEiE8
q5M5IV187I2SHj0DEOl7YEwfdaQN2ZQ+gC7yZiuo8QKBgQC8xNvOqpiz4OVuauFK
2FLITKuvCSIEjaeVQAEDlTh6/XIjOl6yQOhkQ5Shq2XUu+U5A7R/HR8EFPjRdS6o
jVcftO16DNzUNVkXE/cSfwdudKKfXunC4O/KLMH7aTp/R8UvuLDYx2rgL7SKnBWL
XL8LKlYguJpjFb1n9Ey4VIGf6wKBgQC56dUi6wkifNfQqr36e6QehV2n9LXYEudS
H1ZOzmdl5UqEWIeEjaeZ+iwxB1qQzysrxuG76GNvDWBKxC6y7/3iDMwe0yJErMe2
y3PPl5mPdbBSc35cFVH0OnMObx0O9rX461eP6wuOdC2JZtgMClhoKl9YaGWl6dTg
ct/TI7IoRQKBgAu50smgsUGjO71DKGGE1Ieas6N41sMfpYRQ6H83er2xFrD3hEBg
Zc8sn4P3aKjxTwjeoQywW2fegaJJPC++L+ZJT+Mf/2uzMKiNVfCxMV38IsnxlGXF
oT+Wzin23Gpho+8V8u5UXOA2NmnRzkjM5xUjiQMZVIXJKO3XMBjXLwInAoGAcRxM
OYdoMlxkbadEGJ6a9prLpYEkFcdCh9tbBm1byUncsXKvWL1aiDsGIfpIZqw0ptjE
xC+aN5mayUJc6y9Kdvq0QdjRE5lPlO5pkpa4rei7kx6DdYDfMMlS8Sg3NMBzbHO6
/WvSxyP2BI/WyFSO/ykwgzhZB72hb8FUXkBsh0UCgYAikQFEAJmgCDPWO1a4Ts3C
pUEH8ZlPDj77aL/otxdr+UiEOBzmWknlb0Nq+KtEHbq3bk0wQ9aC7lnUT+HgJJ2z
th/mxilgYeWF2CH5DvtZRNIDR0ihuMQe4dDcFyA/cGxrGwYUXX+XTnLT8krGZ1Nv
O6I9o1BVn8DVCy1LaI7e0Q==
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
SampleCCA62gAwIBAgIJAIBi92yKX7vIMA0GCSqGSIb3DQEBCwUAMIGYMQswCQYD
VQQGEwJHQjEOMAwGA1UECAwFSEFOVFMxEDAOBgNVBAcMB0h1cnNsZXkxDDAKBgNV
BAoMA0lCTTEZMBcGA1UECwwQc3BlY3RydW1kaXNjb3ZlcjEZMBcGA1UEAwwQc3Bl
Y3RydW1kaXNjb3ZlcjEjMCEGCSqGSIb3DQEJARYUbWxhd3JlbmNlQHVrLmlibS5j
b20wHhcNMTgxMDE1MjI0OTM4WhcNMzgxMDEwMjI0OTM4WjCBmDELMAkGA1UEBhMC
R0IxDjAMBgNVBAgMBUhBTlRTMRAwDgYDVQQHDAdIdXJzbGV5MQwwCgYDVQQKDANJ
Qk0xGTAXBgNVBAsMEHNwZWN0cnVtZGlzY292ZXIxGTAXBgNVBAMMEHNwZWN0cnVt
ZGlzY292ZXIxIzAhBgkqhkiG9w0BCQEWFG1sYXdyZW5jZUB1ay5pYm0uY29tMIIB
IjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPPx8WO7TxpKbmMmOWd5MwPf
4CenKrgpYmeU+gQbgFIINEPMa9jG2lUetxk36rN4GxUlsvJpYaFIDDyG8or2KrZo
Q7yBagYmVbN6dhrKwH8YlBS0sJnockM8w8kGYc6ENT9dNH9/jaaBZXX9dIlgdjQE
oSOuu1CJgu+vGK2cwFqp/LSmtJDKdgM8/aSa8YTyEUg7HNz+fv2m7DkGyjvTkRem
sXzZCXIzFbvN4boJ7QnZgVDjvunYrTCMJiOautIFaYv00mzhPxXf+dzbkffcR4Hu
qZbPGyAbN+ydwt+8QSJfuDaEQAhHkSKVnCxiwMhaAawOE8/DyPRzzZnHk6AV8wID
AQABo4IBDjCCAQowHQYDVR0OBBYEFJz0+e8N+mU+diD3emP9E5/MAwdxMIHNBgNV
HSMEgcUwgcKAFJz0+e8N+mU+diD3emP9E5/MAwdxoYGepIGbMIGYMQswCQYDVQQG
EwJHQjEOMAwGA1UECAwFSEFOVFMxEDAOBgNVBAcMB0h1cnNsZXkxDDAKBgNVBAoM
A0lCTTEZMBcGA1UECwwQc3BlY3RydW1kaXNjb3ZlcjEZMBcGA1UEAwwQc3BlY3Ry
dW1kaXNjb3ZlcjEjMCEGCSqGSIb3DQEJARYUbWxhd3JlbmNlQHVrLmlibS5jb22C
CQCAYvdsil+7yDAMBgNVHRMEBTADAQH/MAsGA1UdDwQEAwIBBjANBgkqhkiG9w0B
AQsFAAOCAQEAkP3buR6yS5CQ634jisvv+vSTCbYauV8ZNaNoudBBl1QiiWR0SPVn
aR+oPtASZ7Ab3uICdo1/Bf9Wo1LzdTEpnfMmYQOG7C2UNj8TDvFIhCUutkMmO5MB
NDIH7Q/XUqSXqrBq+KkrfEl5MqqPLU8yelEF1L0U7MHLBP3h2fz528hnLBL5rjI+
NbM7yJFIA4nsnVoqUZbWZOGiAqZ0uPCFLC6hyD/DzT2rfHgfiS3Y9sLUdRMhHd8o
j6w8+G8J3B5lhqbAueKcNXS6ZCBIPJtiovZtsaT4AnslpB2GRJoWJQvxJzr1OLCm
```

```
KfRlkCJTnbPKfI34HxOE/k/wpOJSample==
-----END CERTIFICATE-----
```

The response contains the client certificate and the client key, followed by the root certificate. Based on the response output, the organization of the response is as follows:

```
-----BEGIN CERTIFICATE-----
client certificate
-----END CERTIFICATE-----
Bag Attributes
...
Key Attributes:
...
-----BEGIN PRIVATE KEY-----
client key
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
root certificate
-----END CERTIFICATE-----
```

These certificates and key are required for the secure communication connection to the Kafka brokers in IBM Data Cataloging.

Note: Before you connect to the Kafka brokers in IBM Data Cataloging, the application must complete the additional settings for the secure connection.

/policyengine/v1/action_ids: GET

Retrieves and displays a list of actions IDs of all the registered applications.

This API endpoint provides a unique list of action IDs for the registered applications.

The following table displays the roles that can access the REST API endpoint.

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	√	√	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_host>/policyengine/v1/
action_ids -X GET -H 'Accept: application/json'
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example returns a list of action IDs of registered applications that are configured in the system.

Issue the following request in one line:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>
/policyengine/v1/action_ids -X GET -H 'Accept: application/json'
```

The following response is returned:

```
[
    "CONTENTSEARCH",
    "COPY",
    "AUTOTAG",
    "TIER"
]
```

/policyengine/v1/applications/<deployment_name>/schema?action_id=<action_id>: GET

Gets the application schema for a specified application.

This endpoint returns the schema from the application registration message. The schema is used by the UI to dynamically generate policy creation views for the registered application. If the action ID is not specified as a query parameter, the endpoint returns the whole schema. For the specified action ID, a subset of the registration schema is returned.

The following table displays the roles that can access the REST API endpoint.

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓	✓	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_host>/policyengine/v1/
applications/<deployment_name>/schema?action_id=<action_id>  -X GET -H 'Accept: application/json'
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example returns the application schema for a specified application and a specified action ID that is registered on the system.

Issue the following request in one line:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/policyengine/v1/applications/ScaleILM/schema?action_id=MOVE -X GET -H 'Accept: application/json'
```

The following response is returned:

```
{
  "properties": {
    "destination_tier": {
      "description": "Tier name which can be a Spectrum Scale internal pool name or an external pool name like 'tape-archive'
to tier the files to a LTFS Tape library configured on the Spectrum Scale cluster. Internal pool names are mentioned as-is,
while the external pools are mentioned as 'external:tape-archive:pool1@lib1' or 'external:cos' (future use). External pool name
'tape-archive' is an IBM Spectrum Discover defined pool name for the LTFSEE Tape pool configured on the Spectrum Scale
system.",
      "type": "string"
    },
    "source_connection": {
      "description": "Defines platform name of the source data connection.",
      "type": "string"
    }
  },
  "required": [
    "source_connection",
    "destination_tier"
  ],
  "type": "object"
}
```

/policyengine/v1/application_names?action_id=<action_id>: GET

Gets a list of applications that support a specified action ID.

This API endpoint provides a unique list of application names with the specified action ID.

If an action ID is not provided as a query parameter then the endpoint returns all the application names on the system.

The following table displays the roles that can access the REST API endpoint.

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓	✓	X	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_host>/policyengine/v1/
application_names?action_id=<action_id>  -X GET -H 'Accept: application/json'
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example returns a list of application names with the given action id that are configured in the system.

Issue the following request in one line:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>
/policyengine/v1/action_ids -X GET -H 'Accept: application/json'
```

The following response is returned:

```
[
    "WKCConnector",
    "contentsearchagent",
    "ScaleILM"
]
```

/policyengine/v1/applications: DELETE

Deletes an application.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	CollectionAdmin	Data user	Admin	Service user
✓	X	X	X	X

Synopsis of the request URL

```
curl -k -H "Authorization: Bearer <_token_>" -X DELETE
https://<data_cataloging_host>/policyengine/v1/applications/<application_name>
```

Supported request types and response formats

Supported request types:

- DELETE

Examples

Deleting an application involves the following steps.

1. Obtain an auth token by using credentials of the data admin user as shown in the following example:

   ```
   curl -k https://<data_cataloging_host>/auth/v1/token -u "<user_name>:<password>" -v
   ```

   For a valid user, the auth token is returned in the **"X-Auth-Token"** response header.

2. Submit the following request:

   ```
   curl -k https://<data_cataloging_host>/policyengine/v1/applications/<application_name> " -X DELETE -H "Authorization:
   Bearer <_token_> "
   ```

   Response:

   ```
   204 No Content
   ```

RBAC management by using APIs

The IBM Data Cataloging resource-based access control (RBAC) is a REST API service that enables role-based access to the IBM Data Cataloging services. This service uses OpenStack Keystone as a backend for providing Identity and Access Management (IAM) across multiple domains that are attached to the IBM Data Cataloging.

The authentication service uses a default user with user name *sdadmin* and password *Passw0rd* in the domain named *default*. This user has the administrative role and can be used to do the following actions:

- This user can create other users and user groups.
- This user can register new domains with the authentication service.
- This user can create projects.
- This user can assign roles to the users at project or domain level.

The following are the predefined user roles with the corresponding access levels:

admin
: Default user role created by the system. Users with this role can create other users, projects, domains, and assign roles. Users with this role cannot see metadata records.

dataadmin
: The users with this role can see all metadata records across projects.

collectionadmin

The users with this role can access metadata that is collected. But, metadata access is restricted to the records that are associated with the collections to which the user has the Collection Admin or Data User role assigned.
Important: The Collection Admin role is available as a technology preview in the 2.0.1.1 release. For limitations on the usage of the Collection Admin role, see the *IBM Spectrum® Discover Release Notes*.

datauser
: This role is ideal for a researcher or data scientist. Users with this role can see records that are associated with projects to which they belong.

serviceuser
: This user role intended for service personnel. Users with this role have read-only access to the system logs.

IBM Data Cataloging integrates with the enterprise LDAP connected to IBM Spectrum Scale. Using the authentication service APIs, admin users can add an LDAP domain definition to IBM Data Cataloging. The users and groups from the registered LDAP domain are automatically imported into IBM Data Cataloging and the administrators can add these users and groups to different projects while they assign them the Data User role.

Admin users can also assign the Data Admin role to some of the users and user groups to give access to the entire IBM Data Cataloging index for searches and policies.

The authentication API service endpoint has the following basic structure: `https://<host address>/auth/v1/<endpoint>`

For example, the following endpoint gets authentication token for the users: `curl -k -u <user>:<pass> https://<host address>/auth/v1/token`

- **Managing tokens**
  The users need to get an authentication token from the authentication server by using their user name and password. Then, they need to use that token to access the services offered by the IBM Data Cataloging system.
- **Managing users**
  All the create, read, update, and delete operations are allowed on the *default* domain. But the read-only (GET) access is allowed for the users and groups that are imported from the registered Lightweight Directory Access Protocol (LDAP) or Cloud Storage Object domains.
- **Managing user roles**
  The authorization or access privileges of users and user groups are defined with the help of user roles.
- **Managing user groups**
  You can create user groups for better manageability of users. Defining user groups also provides flexibility of assigning specific roles to a group of users rather than assigning the roles individually.
- **Managing collections**
  You can create and manage collections by using REST APIs.
- **Managing domains**
  You can create and manage user domains by using REST APIs.

Managing tokens

The users need to get an authentication token from the authentication server by using their user name and password. Then, they need to use that token to access the services offered by the IBM Data Cataloging system.

For more information on the authentication process, see Authentication process.

The following endpoint is available to manage tokens:

- **/auth/v1/token: GET**
  Gets authentication token for the user.

/auth/v1/token: GET

Gets authentication token for the user.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓	✓	✓	✓

Synopsis of the request URL

Users from an external LDAP or COS domain must include domain name in the user name as *<domain>/<user>* to get an authentication token.

```
curl -k -u ldap/user1:pass https://<data_cataloging_host>/auth/v1/token -v
```

```
curl -k -u cos/user1:pass https://<data_cataloging_host>/auth/v1/token -v
```

```
curl -k -u cos/<access_key_id>:<secret_key> https://<data_cataloging_host>/auth/v1/token -v
```

Without specifying domain name as part of user name:

```
curl -k -u user1:pass https://<data_cataloging_host>/auth/v1/token -v
```

Note: When using curl, add the -v parameter to see the response headers that contain the token.

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Default response data is in CSV format. For results in JSON format, use the "Accept: application/json" header.

Examples

The following example shows a request and the corresponding response.

Request that contains domain as part of the user name:

```
curl -k -u ldap/user1:pass https://<data_cataloging_host>/auth/v1/token -v
```

Request that does not contain domain as part of the user name:

```
curl -k -u user1:pass https://<data_cataloging_host>/auth/v1/token -v
```

Response:

```
< HTTP/1.0 200 OK
< X-Auth-Token: gAAAAABbUdgLOLk67Zk_XQO0bbJt_qe4suz449m7XLM-D6e0jUzDxBW574L38y5xF5y3wc-
Xq0Bp43uQH13QN-wTsioPoOfOlbyrN5avBag2iHUkCOgXyA46TvY5LdGu46LEs-gO-
qidQgUCg5im4QF3Zvw5enCTvQd2N1bAg186CVOc8Qk5o1ldJGd51nL3ELts0LsVg9F4oPhv9HixwZflhlCsuPg1Gw
< Content-Type: text/html; charset=utf-8
< Content-Length: 0
< Server: Werkzeug/0.14.1 Python/2.7.5
< Date: Fri, 20 Jul 2018 12:39:39 GMT
```

You need to use the token that is obtained in the previous step to access the services offered by IBM Data Cataloging.

The following example shows how to use the token that is obtained to get the list of from the policy engine:

```
curl -v -k -H 'Authorization: Bearer gAAAAABbUdgLOLk67Zk_XQO0bbJt_qe4suz449m7XLM-
D6e0jUzDxBW574L38y5xF5y3wc-Xq0Bp43uQH13QN-wTsioPoOfOlbyrN5avBag2iHUkCOgXyA46TvY5LdGu46LEs-
gO-qidQgUCg5im4QF3Zvw5enCTvQd2N1bAg186CVOc8Qk5o1ldJGd51nL3ELts0LsVg9F4oPhv9HixwZflhlCsuPg1Gw'
https://<data_cataloging_host>/policyengine/v1/policies
```

Managing users

All the create, read, update, and delete operations are allowed on the *default* domain. But the read-only (GET) access is allowed for the users and groups that are imported from the registered Lightweight Directory Access Protocol (LDAP) or Cloud Storage Object domains.

- **/auth/v1/users: GET**
  Gets the details of the users that are created in the system.
- **/auth/v1/users: POST**
  Creates a user.
- **/auth/v1/users/<user_ID>/password: POST**
  Changes the user password.
- **/auth/v1/users/<user ID>: GET**
  Gets the details of a specific user.
- **/auth/v1/users/<user_ID>: PATCH**
  Updates an existing user.
- **/auth/v1/users/<user_ID>: DELETE**
  Deletes a specific user.
- **/auth/v1/users/<user_ID>/groups: GET**
  Gets the details of the groups to which a specific user belongs.
- **/auth/v1/users/<user_ID>/collections: GET**
  Gets the list of collections that are assigned to a user.
- **/auth/v1/users/<user_ID>/roles: GET**
  Gets the list of roles that are assigned to a user.
- **/auth/v1/users/<user_ID>/role_assignment: GET**
  Returns a list of the roles and associated collections assigned to the user where applicable. This does not include any role assignments inherited from groups that the user belongs to.
- **/auth/v1/users/users_summary: GET**
  Gets the list of users and their domain details.

/auth/v1/users: GET

Gets the details of the users that are created in the system.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√[1]	√[1]	√	√	√[1]

[1]The specified user ID must be the same as the requesting user ID. Otherwise the command fails with error 403, "Not authorized".

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of the uses who are created in the system.

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users
```

Response:


```
(200 OK):
{
    "users": [
        {
            "domain_id": "default",
            "enabled": true,
            "id": "2e905adec130453f888a425dfc038f9c",
            "name": "sdadmin",
            "options": {},
            "password_expires_at": null
        },
        {
            "domain_id": "default",
            "enabled": true,
            "id": "4ee5b47f439640d29b6fac7253a64290",
            "name": "kris",
            "options": {},
            "password_expires_at": null
        },
        {
            "domain_id": "default",
            "enabled": true,
            "id": "a8dbddcd66b746b9958c34fd5a28855c",
            "name": "dataadmin-user",
            "options": {},
            "password_expires_at": null
        },
        {
            "domain_id": "default",
            "enabled": true,
            "id": "f731680239d7457aa2bad27b276d01f4",
            "name": "kris-tsv-txt",
            "options": {},
            "password_expires_at": null
        },
        {
            "domain_id": "bbd9f2993849402490cefc015013b6e9",
            "id": "b7be4be4f78f6c371583cf4b3617e47900477f6d09f732c2493a22b79d99a2a3",
            "name": "dataadmin",
            "options": {},
            "password_expires_at": null
        },
        {
            "domain_id": "bbd9f2993849402490cefc015013b6e9",
            "id": "0337d1845a17fa29ff6ba83c2fb62870273030dfb4d10de8dfab2e9fbe68aeef",
            "name": "user1",
            "options": {},
            "password_expires_at": null
        },
        {
            "domain_id": "bbd9f2993849402490cefc015013b6e9",
            "id": "d523a89af4e3c7042b564d241dcb058cd892030b63d00091128ba1d1d98a5d73",
            "name": "user2",
            "options": {},
            "password_expires_at": null
        }
    ]
}
```


/auth/v1/users: POST

Creates a user.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X POST
https://<data_cataloging_host>/auth/v1/users -d '<json details of user>'
```

The details of a user can be the following:

- Name
- Password
- Email
- Description
- Project
- Default project

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

The following example shows how to create a user.

Request:

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X POST
https://<data_cataloging_host>/auth/v1/users -d '{"name":"testuser","password":"testpass",
"email":"testuser@ibm.com"}'
```

Response:

```
201 Created
```

/auth/v1/users/<user_ID>/password: POST

Changes the user password.

The table shows the roles that can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓	✓	✓	✓

Synopsis of the request URL

```
curl -k -H 'Content-Type: application/json' -X POST
https://<data_cataloging_host>/auth/v1/users/user_id/password -d'<json details of user>'
```

Required Parameters in the JSON payload:

- password - new password
- original_password - old/current password

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

The following example shows how to change a user password.

Request:

```
curl -k -H 'Content-Type: application/json' -X POST
https://<data_cataloging_host>/auth/v1/users/a26f98ab046449bebce70c4832b22ac2/password -X POST-H"Content-Type:application/json"
-d '{"password":"NewPassw0rd","original_password":"Passw0rd"}'
```

Response:

```
204 No Content
```

/auth/v1/users/<user ID>: GET

Gets the details of a specific user.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√[1]	√[1]	√	√	√[1]
[1]The specified user ID must be the same as the requesting user ID. Otherwise the command fails with error 403, "Not authorized". An exception is if the user ID is "`me`", the result is the details of the current user.				

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/<user_ID>
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of a specific user.

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/4ee5b47f439640d29b6fac7253a64290
```

Response:

```
200 OK:
{
    "domain_id": "default",
    "email": "testuser@ibm.com",
    "enabled": true,
    "id": "5b3cd6af1c38479aa3a8cb220230c651",
    "name": "testuser",
    "options": {},
    "password_expires_at": null
}
```

The following special case returns the current user's details.

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/me
```

/auth/v1/users/<user_ID>: PATCH

Updates an existing user.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	√	X

Synopsis of the request URL

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/users/<user_ID> -X PATCH -d '<json details to update>
```

Supported request types and response formats

You can update the following attributes of a user:

- Name
- Password
- Email
- Description
- Project
- Default project

Supported request types:

- PATCH

Supported response formats:

- JSON

Examples

The following example shows how to update an existing user.

Request:

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/users/4ee5b47f439640d29b6fac7253a64290
-X PATCH -d '{"name": "testuser-updated", "password": "newpassword"}'
```

Response:

```
200 OK
```

/auth/v1/users/<user_ID>: DELETE

Deletes a specific user.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/<user_ID> -X DELETE
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

The following example shows how to delete a user.

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/4ee5b47f439640d29b6fac7253a64290
```

Response:

```
204 No Content
```

/auth/v1/users/<user_ID>/groups: GET

Gets the details of the groups to which a specific user belongs.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓[1]	✓[1]	✓	✓	✓[1]
[1]The specified user ID must be the same as the requesting user ID. Otherwise the command fails with error 403, "Not authorized".				

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/<user_ID>/groups
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of the groups to which a specific user belongs.

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/
4ee5b47f439640d29b6fac7253a64290/groups
```

Response:

```
{
    "groups": [
        {
            "name": "test-group1",
            "description": "Test Group 1",
            "domain": "default"
        },
        {
            "name": "test-group2",
            "description": "Test Group 2",
            "domain": "default"
        }
    ]
}
```

/auth/v1/users/<user_ID>/collections: GET

Gets the list of collections that are assigned to a user.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√[1]	√[1]	√	√	√[1]

[1]The specified user ID must be the same as the requesting user ID. Otherwise the command fails with error 403, "Not authorized".

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/<user_ID>/collections
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of the collections to which the user is assigned.

Request:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/users/4ee5b47f439640d29b6fac7253a64290/collections
```

Response:

```
(200 OK):
{
    "collections": [
        {
            "description": "",
```

```
            "domain_id": "default",
            "enabled": true,
            "id": "1d657422853c4c33bde953837c709655",
            "is_domain": false,
            "name": "JPGProj",
            "parent_id": "default",
            "tags": []
        },
        {
            "description": "",
            "domain_id": "default",
            "enabled": true,
            "id": "28bafe9031af4aa7a637b00aa436fb01",
            "is_domain": false,
            "name": "XMLProj",
            "parent_id": "default",
            "tags": []
        },
        {
            "description": "Bootstrap collection for initializing the cloud.",
            "domain_id": "default",
            "enabled": true,
            "id": "c64cafe7f69349d0ba79a8fe931be8d2",
            "is_domain": false,
            "name": "spectrum-discover",
            "parent_id": "default",
            "tags": []
        }
    ]
}
```

/auth/v1/users/<user_ID>/roles: GET

Gets the list of roles that are assigned to a user.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√[1]	√[1]	√	√	√[1]

[1]The specified user ID must be the same as the requesting user ID. Otherwise the command fails with error 403, "Not authorized".

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/<user_ID>/roles
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the list of roles that are assigned to a user.

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/4ee5b47f439640d29b6fac7253a64290/roles
```

Response:

```
(200 OK):
{
    "roles": [
        {
            "domain_id": null,
            "id": "4a5415cb9cc5460aafe12a6f6206448e",
            "name": "datauser"
        }
    ]
}
```

/auth/v1/users/<user_ID>/role_assignment: GET

Returns a list of the roles and associated collections assigned to the user where applicable. This does not include any role assignments inherited from groups that the user belongs to.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	✓	✓	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/<user_id>/role_assignments
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of the roles that is assigned to a specific user:

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/
4ee5b47f439640d29b6fac7253a64290/role_assignments
```

Response:

```
Returns 200 OK:
{
    "roles":[
        {
            "collections":[
                {
                    "description":"Bootstrap project for initializing the cloud.",
                    "domain_id":"default",
                    "enabled":true,
                    "id":"09dd9136eaa6416784d6c5bdf3fa2f78",
                    "is_domain":false,
                    "name":"data-cataloging",
                    "parent_id":"default",
                    "tags":[]
                }
            ],
            "domain_id":"domain_id",
            "id":"5f365cf289d14c93a986b8bea976f92b",
            "name":"collectionadmin"
        }
    ]
}
```

/auth/v1/users/users_summary: GET

Gets the list of users and their domain details.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓[1]	✓	✓	✓[1]

[1] The specified user ID must be the same as the requesting user ID. Otherwise the command fails with error 403, "Not authorized". Data users can access this API to retrieve only those user details they are entitled to view.

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/users_summary
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the user details.

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/users/users_summary
```

Response:

```
(200 OK):
 {
    "users": [
        {
               "domain_id": "default",
               "domain_name": "Default",
               "enabled": true,
               "id": "7d040d976f924d60b5f39f0648950d1b",
               "name": "sdadmin",
               "options": {},
               "password_expires_at": null
        },
        {
               "domain_id": "a69001e854174dc1a4e0c98f712744bb",
               "domain_name": "lldap",
               "id": "1e8155daee6f021008a09aca09e1473d823ca7128d16ba25b3a10a7c1aa3433e",
               "name": "user2_NoGroup",
               "options": {},
               "password_expires_at": null
        },
        {
               "domain_id": "a69001e854174dc1a4e0c98f712744bb",
               "domain_name": "lldap",
               "id": "38f5434fe9d12fcb3278deb29fb104441b87d040f5774bd545a2d1d0c7adaa7f",
               "name": "user5_SystemAdmin",
               "options": {},
               "password_expires_at": null
        }
    ]
}
```

Managing user roles

The authorization or access privileges of users and user groups are defined with the help of user roles.

The following are the predefined user roles with the corresponding access levels:

admin
: Users with this role can create other users, projects, domains, and assign roles. Users with this role cannot see metadata records. This role is a default user role that is created by the system.

dataadmin
: Users with this role can see all metadata records across projects.

collectionadmin
: Users with this role can access metadata that is collected. However, metadata access is restricted to the records that are associated with the collections to which the user has the Collection Admin or Data User role assigned.
Important: The Collection Admin role is available as a technology preview in the 2.0.1.1 release. For limitations on the usage of the Collection Admin role, see the *IBM Spectrum® Discover Release Notes*.

datauser
: Users with this role can see records that are associated with projects to which they belong. This role is ideal for a researcher or data scientist.

serviceuser
: Users with this role have read-only access to the system logs. This user role intended for service personnel.

- **/auth/v1/roles/<role_ID>: PUT**
  Assigns a role to users or user groups.
- **/auth/v1/roles/<role_ID >: GET**
  Gets the details of a specific user role.
- **/auth/v1/roles: GET**
  Gets the details of user roles.
- **/auth/v1/roles/<role_ID>: DELETE**
  Removes the role that is assigned to a user, group, collection, or domain.

/auth/v1/roles/<role_ID>: PUT

Assigns a role to users or user groups.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	✓	✓	X

The following roles can be assigned to a user:

admin
: Default user role created by the system. Users with this role can create other users, projects, domains, and assign roles. Users with this role cannot see metadata records.

dataadmin
: The users with this role can see all metadata records across projects.

datauser
: This role is ideal for a researcher or data scientist. Users with this role can see records that are associated with projects to which they belong.

serviceuser
: This user role intended for service personnel. Users with this role have read only access to the system logs.

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' -X PUT https://<data_cataloging_host>/auth/v1/roles/<role_ID> -d '<details of where role is to be assigned>'
```

You can specify the following details in the **roles** endpoint to assign roles to the corresponding component. An example follows this list.

user_id
: The user ID to which the role must be assigned.

group_id
: The group to which the role must be assigned.

project_id
: The project to which the role must be assigned.

domain_id
: The domain to which the role must be assigned.

The following examples illustrate how to specify these fields to assign a role to a domain or collection:

```
{
    "user_id": <string> [optional],
    "group_id": <string> [optional],
    "collection_id": <string> [optional],
    "domain_id": <string> [optional]
}
example [assign role to user in default domain]
{
    "user_id": "5b3cd6af1c38479aa3a8cb220230c651"
}
example [assign role to user in specific domain]
{
    "user_id": "5b3cd6af1c38479aa3a8cb220230c651",
    "domain_id": "4bfa9a9f71154dbeb554261a1c8d9a53"
}
example [assign role to user in specific collection]
{
    "user_id": "5b3cd6af1c38479aa3a8cb220230c651",
    "collection_id": "3b8c501b173c4ebcb99f322a4f7cf60e"
}
```

Supported request types and response formats

Supported request types:

- PUT

Supported response formats:

- JSON

Examples

Example 1: Assigning a role to a specific user.

Request:

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X PUT
https://<data_cataloging_host>/auth/v1/roles/4a5415cb9cc5460aafe12a6f6206448e
-d '{"user_id": "5b3cd6af1c38479aa3a8cb220230c651"}'
```

Response:

```
200 OK
```

Example 2: Assigning a role to a group.

Request:

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X PUT
https://<data_cataloging_host>/auth/v1/roles/4a5415cb9cc5460aafe12a6f6206448e
-d '{"group_id":  "6c4de7af1c38479aa3a8cb220230d762"}'
```

Response:

```
200 OK
```

Example 3: Assigning role for a user in a collection.

Request:

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X PUT
https://<data_cataloging_host>/auth/v1/roles/4a5415cb9cc5460aafe12a6f6206448e
-d '{"user_id": "5b3cd6af1c38479aa3a8cb220230c651", "domain_id": "af1c38479a5b3cd6c651a3a8cb220230"}'
```

Response:

```
200 OK
```

/auth/v1/roles/<role_ID >: GET

Gets the details of a specific user role.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√[1]	√[1]	√[1]	√	√[1]
[1]The specified role must be one that is assigned to the requesting user ID. Otherwise the command fails with error 403, "Not authorized".				

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/roles/<role_ID>
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of a specific user role.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/roles/4a5415cb9cc5460aafe12a6f6206448e
```

Response:

```
200 (OK):
{
    "domain_id": null,
    "id": "4a5415cb9cc5460aafe12a6f6206448e",
    "name": "datauser"
}
```

/auth/v1/roles: GET

Gets the details of user roles.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√[1]	√[1]	√[1]	√	√[1]
[1]The specified role must be one that is assigned to the requesting user ID. Otherwise the command fails with error 403, "Not authorized".				

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/roles
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of user roles.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/roles
```

Response:


```
{
  "roles": [
    {
      "domain_id": null,
      "id": "0ce17850c81a4e5da06ecd93f8ea0393",
      "name": "admin"
    },
    {
      "domain_id": null,
      "id": "383d4b371f244a2a8529a35b665b5c01",
      "name": "datauser"
    },
    {
      "domain_id": null,
      "id": "7d9f6d213b6d4317a395bbf248fdd9b6",
      "name": "collectionadmin"
    },
    {
      "domain_id": null,
      "id": "103bc9b8f1864c19af5663ec357f8d51",
      "name": "dataadmin"
    },
    {
      "domain_id": null,
      "id": "8388af928ce546bea7a0cd804a427b7c",
      "name": "serviceuser"
    }
  ]
}
```


/auth/v1/roles/<role_ID>: DELETE

Removes the role that is assigned to a user, group, collection, or domain.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	✓	✓	X

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' -X DELETE https://<data_cataloging_host>/auth/v1/
roles/<role_ID> -d '<the component for which the role must be revoked >'
```

You can specify one of the following items for which you need to revoke the role:

- user_id
- group_id
- collection_id
- domain_id

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

Example 1: Revoking role from a user.

Request:

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X DELETE
https://<data_cataloging_host>/auth/v1/roles/4a5415cb9cc5460aafe12a6f6206448e
```

```
-d '{"user_id": "5b3cd6af1c38479aa3a8cb220230c651"}'
```

Response:

```
204 No Content
```

Example 2: Revoking role from a group.

Request:

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X DELETE
https://<data_cataloging_host>/auth/v1/roles/4a5415cb9cc5460aafe12a6f6206448e
-d '{"group_id": "6c4de7af1c38479aa3a8cb220230d762"}'
```

Response:

```
204 No Content
```

Example 3: Revoking role for a user of a collection.

Request:

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X DELET
https://<data_cataloging_host>/auth/v1/roles/4a5415cb9cc5460aafe12a6f6206448e
-d '{"user_id": "5b3cd6af1c38479aa3a8cb220230c651", "collection_id": "c38479aa35b3cd6af1a8cb220230c651"}'
```

Response:

```
204 No Content
```

Example 4: Revoking role for a group of a domain.

Request:

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X DELETE
https://<data_cataloging_host>/auth/v1/roles/4a5415cb9cc5460aafe12a6f6206448e
-d '{"user_id": "5b3cd6af1c38479aa3a8cb220230c651", "domain_id": "af1c38479a5b3cd6c651a3a8cb220230"}'
```

Response:

```
204 No Content
```

Managing user groups

You can create user groups for better manageability of users. Defining user groups also provides flexibility of assigning specific roles to a group of users rather than assigning the roles individually.

- **/auth/v1/groups: GET**
  Gets the list of groups that are created in the system.
- **/auth/v1/groups: POST**
  Creates a group in the system.
- **/auth/v1/groups/<group ID>: GET**
  Gets the details of a specific group.
- **/auth/v1/groups/<group_ID>: PATCH**
  Updates the attributes of an existing group.
- **/auth/v1/groups/<group_ID>: DELETE**
  Deletes a specific user group.
- **/auth/v1/groups/<group_ID>/collections: GET**
  Gets the details of the collections that are assigned in a group.
- **/auth/v1/groups/<group_ID>/roles: GET**
  Gets the details of the user roles that are assigned to a group.
- **/auth/v1/groups/<group_id>/role_assignments: GET**
  Returns a list of the roles and associated collections assigned to the group where applicable.
- **/auth/v1/groups/<group_ID>/users: GET**
  Gets the details of the users who are part of a specific group.
- **/auth/v1/groups/<group_ID>/user/<user_ID>: DELETE**
  Removes a specific user from a group.
- **/auth/v1/groups/<group_ID>/user/<user_ID>: PUT**
  Assigns a user to a group.
- **/auth/v1/groups/groups_summary: GET**
  Gets the list of groups and their domain details.

/auth/v1/groups: GET

Gets the list of groups that are created in the system.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√[1]	√[1]	√	√	√[1]

Data admin	Data user	Collection Admin	Admin	Service user
[1]The specified group must be one that is assigned to the requesting user ID. Otherwise the command fails with error 403, "Not authorized".				

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of the uses that are created in the system.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups
```

Response:

```
(200 OK):
{
    "groups": [
        {
"id": "a51f84bfeb034b12a9a362109d78bec3",
            "name": "test-group1",
            "description": "Test Group 1",
            "domain": "default"
        },
        {
"id": "b62g95bfeb034b12a9a362109d89cfd4",
            "name": "test-group2",
            "description": "Test Group 2",
            "domain": "default"
        }
    ]
}
```

/auth/v1/groups: POST

Creates a group in the system.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	✓	✓	X

Synopsis of the request URL

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X POST
https://<data_cataloging_host>/auth/v1/groups -d '<json details of group>'
```

You can specify the following details of a group:

- "name": Name of the group.
- "description": A meaningful description for the group.
- "domain_id": The domain to which the group belongs.

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

The following example shows how to create a group.

Request:

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/groups -d '{"name": "test-group"}'
```

Response:

```
201 Created
```

/auth/v1/groups/<group ID>: GET

Gets the details of a specific group.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√[1]	√[1]	√[1]	√	√[1]
[1]The specified group must be one that is assigned to the requesting user ID. Otherwise the command fails with error 403, "Not authorized".				

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/<group_ID>
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of a specific group.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/a51f84bfeb034b12a9a362109d78bec3
```

Response:

```
(200 OK):
{
    "id": "a51f84bfeb034b12a9a362109d78bec3",
    "name": "test-group1",
    "description": "Test Group 1",
    "domain": "default"
}
```

/auth/v1/groups/<group_ID>: PATCH

Updates the attributes of an existing group.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	√	X

Synopsis of the request URL

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/groups/<group_ID> -X PATCH -d '<json details to update>'
```

Supported request types and response formats

You can update the following attributes of a group.

- Name
- Description
- Domain ID

You can update one or more attributes in a single call.
Supported request types:

- PATCH

Supported response formats:

- JSON

Examples

The following example shows how to update the attributes of a group:

Request:

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/groups/a51f84bfeb034b12a9a362109d78bec3 -X PATCH -d '{"name": "AdminGroup",
"description":"Verifies installations", "domain_id": "Default"}'
```

Response:

```
200 OK
```

/auth/v1/groups/<group_ID>: DELETE

Deletes a specific user group.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/<group_id> -X DELETE
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

The following example shows how to delete a group.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/a51f84bfeb034b12a9a362109d78bec3 -X
DELETE
```

Response:

```
204 No Content
```

/auth/v1/groups/<group_ID>/collections: GET

Gets the details of the collections that are assigned in a group.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/<group_ID>/collections
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of the collections assigned within a user group.

Request:

```
curl -k  -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/groups/a51f84bfeb034b12a9a362109d78bec3/collections
```

Response:

```
((200 OK):
[
    {
        "description": "This is a test collection",
        "domain_id": "default",
        "enabled": true,
        "id": "51d5bf6c7c4e4fd3b35c53ca91d95705",
        "is_domain": false,
        "name": "test-collection",
        "parent_id": "default",
        "tags": [
            "tag1",
            "tag2"
        ]
    }
]
```

/auth/v1/groups/<group_ID>/roles: GET

Gets the details of the user roles that are assigned to a group.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/<group_ID>/roles
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of the user roles that are assigned to a user group.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/
a51f84bfeb034b12a9a362109d78bec3/roles
```

Response:

```
(200 OK):
[
    {
        "domain_id": "default",
               "id": "1bdcad2ec7f2414793e71b8b505515b1",
               "name": "datauser"
    }
]
```

/auth/v1/groups/<group_id>/role_assignments: GET

Returns a list of the roles and associated collections assigned to the group where applicable.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	✓	✓	X

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/<group_id>/role_assignments
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of the roles that is assigned to a specific group:

Request:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/groups/a51f84bfeb034b12a9a362109d78bec3/role_assignments
```

Response:

```
Returns 200 OK:
{
    "roles":[
        {
            "collections":[
                {
                    "description":"Bootstrap project for initializing the cloud.",
                    "domain_id":"default",
                    "enabled":true,
                    "id":"09dd9136eaa6416784d6c5bdf3fa2f78",
                    "is_domain":false,
                    "name":"spectrum-discover",
                    "parent_id":"default",
                    "tags":[
                    ]
                },
                {
                    "description":"kutle",
                    "domain_id":"default",
                    "enabled":true,
                    "id":"40b0ea6fca7a4bafa38581cb508aa373",
                    "is_domain":false,
                    "name":"some-collection",
                    "parent_id":"default",
                    "tags":[
                    ]
                }
            ],
            "domain_id":null,
            "id":"5f365cf289d14c93a986b8bea976f92b",
            "name":"collectionadmin"
        }
    ]
}
```

/auth/v1/groups/<group_ID>/users: GET

Gets the details of the users who are part of a specific group.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/<group_ID>/users
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of the users that are part of a user group.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/
a51f84bfeb034b12a9a362109d78bec3/users
```

Response:

```
(200 OK):
[
    {
        "domain_id": "default",
               "email": "testuser@ibm.com",
               "enabled": true,
               "id": "dfdb2cadca0d4e9da26e6c40e5cadbd5",
               "name": "datauseruser",
               "options": {},
               "password_expires_at": null
    },
    {
        "domain_id": "default",
               "email": "testuser@ibm.com",
               "enabled": true,
               "id": "106999c5eb7f4148932be6d24b6b772c",
               "name": "test-datauser-1",
               "options": {},
               "password_expires_at": null
    }
]
```

/auth/v1/groups/<group_ID>/user/<user_ID>: DELETE

Removes a specific user from a group.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/<group_id>/user/<user_ID> -X DELETE
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

The following example shows how to remove a user from a group.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/
a51f84bfeb034b12a9a362109d78bec3/user/4ee5b47f439640d29b6fac7253a64290 -X DELETE
```

Response:

```
204 No Content
```

/auth/v1/groups/<group_ID>/user/<user_ID>: PUT

Assigns a user to a group.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>
https://<data_cataloging_host>/auth/v1/groups/<group_ID>/user/<user_ID> -X PUT
```

Examples

The following example shows how to assign a user to a group:

Request:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/groups/a51f84bfeb034b12a9a362109d78bec3/user/4ee5b47f439640d29b6fac7253a64290  -X PUT
-H "Content-Type: application/json"
```

Response:

`200` if the commend is successful.

/auth/v1/groups/groups_summary: GET

Gets the list of groups and their domain details.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓[1]	✓	✓	✓[1]
[1] The specified user ID must be the same as the requesting user ID. Otherwise the command fails with error 403, "Not authorized". Data users can access this API to retrieve details only for the groups to which they belong.				

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/groups_summary
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the domain names to which the group is associated.

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/groups/groups_summary
```

Response:

```
(200 OK):
{
        "groups": [
                {
                        "description": "",
                        "domain_id": "default",
                        "domain_name": "Default",
                        "id": "86f7a42ea54d4659bb60fc6d5c620f2d",
                        "name": "bugfix5"
                },
                {
                        "domain_id": "a69001e854174dc1a4e0c98f712744bb",
                        "domain_name": "lldap",
                        "id": "d3b1ae0d1b5efb0c064797bf8239165cce32aaf3c057151279198a780bc8cf0b",
                        "name": "Sales"
                },
                {
                        "domain_id": "a69001e854174dc1a4e0c98f712744bb",
                        "domain_name": "lldap",
                        "id": "18326a0d724b4ffaa26a228fc88b6852198a1d10a3a51bc732632afe4d112dc9",
```

```
                        "name": "HumanResources"
                }
        ]
}
```

Managing collections

You can create and manage collections by using REST APIs.

- **/auth/v1/collections: GET**
  Gets the details of the collections that are available in the system.
- **/auth/v1/collections: POST**
  Creates a collection.
- **/auth/v1/collections/<collection_ID>: GET**
  Gets the details of a specific collection.
- **/auth/v1/collections/<collection_ID>: PATCH**
  Updates the attributes of an existing collection.
- **/auth/v1/collections/<collection_ID>: DELETE**
  Deletes a specific collection.
- **/auth/v1/collections/<collection_ID>/groups: GET**
  Gets a list of the user groups that can access a collection.
- **/auth/v1/collections/<collection_ID>/users: GET**
  Gets a list of the users who can access a collection.
- **/auth/v1/collections/<collection_id>/role_assignments: GET**
  Returns a list of the users and groups and their roles that are assigned to the collection.

/auth/v1/collections: GET

Gets the details of the collections that are available in the system.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	√[1]	√[1]	√	X

[1] The list that is returned comprises the set of collections that are assigned to the requesting User ID. If no collections are assigned, then an empty response is returned.

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/collections
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of collections that are available in the system.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/collections
```

Response:

```
(200 OK):
{
    "collections": [
        {
            "description": "Bootstrap project for initializing the cloud.",
            "domain_id": "default",
            "enabled": true,
            "id": "dfd993527cb34b0bbcd700fab121264e",
            "is_domain": false,
            "name": "spectrum-discover",
            "parent_id": "default",
            "tags": []
        },
        {
            "description": "This is a test collection",
            "domain_id": "default",
            "enabled": true,
```

```
            "id": "51d5bf6c7c4e4fd3b35c53ca91d95705",
            "is_domain": false,
            "name": "test-collection",
            "parent_id": "default",
            "tags": [
                "tag1",
                "tag2"
            ]
        }
    ]
}
```

/auth/v1/collections: POST

Creates a collection.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X POST
https://<data_cataloging_host>/auth/v1/collections -d '{details of the collection}'
```

The details that can be specified are:

- Name
- Description
- Tags
  Important: IBM Data Cataloging does not use collection tags. These collection tags are not related to the IBM Data Cataloging metadata tags.

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

The following example shows how to create a collection.

Request:

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/collections -d '{"name": "test-collection", "description": "Test Collection 1","tags":
["tag1", "tag2"]}'
```

Response:

```
201 Created
```

/auth/v1/collections/<collection_ID>: GET

Gets the details of a specific collection.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	✓[1]	✓[1]	✓	X

[1] The specified collection must be the one that is assigned to the requesting user ID. If no collection is assigned, then the command fails with error 403, "Not authorized".

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/collections/<collection_ID>
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the details of a specific collection.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/collections/51d5bf6c7c4e4fd3b35c53ca91d95705
```

Response:

```
(200 OK):
{
    "description": "This is a test collection",
    "domain_id": "default",
    "enabled": true,
    "id": "51d5bf6c7c4e4fd3b35c53ca91d95705",
    "is_domain": false,
    "name": "test-collection",
    "parent_id": "default",
    "tags": [
        "tag1",
        "tag2"
    ]
}
```

/auth/v1/collections/<collection_ID>: PATCH

Updates the attributes of an existing collection.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	✓	✓	X

Synopsis of the request URL

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/collections/<collection_ID> -X PATCH
-d '<json_details_to_update>'
```

Supported request types and response formats

You can update the following attributes of a collection.

- Name
- Description
- Tags

You can update one or more attributes in a single call.
Supported request types:

- PATCH

Supported response formats:

- JSON

Examples

The following example shows how to update the attributes of a collection:

Request:

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/collections/a51f84bfeb034b12a9a362109d78bec3
-X PATCH -d '{"name": "General", "description":"ThirdQuarter", "tags": {"ProjectNumber":"1404", "DepartmentCode":"H8AC"}'
```

Response:

```
200 OK
```

/auth/v1/collections/<collection_ID>: DELETE

Deletes a specific collection.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X DELETE
https://<data_cataloging_host>/auth/v1/collections/<collection_ID>
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

The following example shows how to delete a specific collection.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/collections/dfd993527cb34b0bbcd700fab121264e
-X DELETE
```

Response:

```
204 No Content
```

/auth/v1/collections/<collection_ID>/groups: GET

Gets a list of the user groups that can access a collection.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	[1]✓	✓	X

[1] The specified collection must be one that is assigned to the requesting user ID. If no collection is assigned, then the command fails with error 403, "Not authorized".

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/collections/<collection_ID>/groups
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get details of user groups that belong to a specific collection.

Request:

```
curl -k  -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/collections/a51f84bfeb034b12a9a362109d78bec3/groups
```

Response:

```
(200 OK):
[
    {
        "description": "",
        "domain_id": "default",
        "id": "d1c1f64d62df4585ad8c76b345e0a651",
        "name": "test-group-1"
```

```
    }
]
```

/auth/v1/collections/<collection_ID>/users: GET

Gets a list of the users who can access a collection.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	X	[1]√	√	X
[1] The specified collection must be one that is assigned to the requesting user ID. If no collection is assigned, then the command fails with error 403, "Not authorized".				

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/collections/<collection_ID>/users
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get details of users who belong to a specific collection.

Request:

```
curl -k  -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/collections/a51f84bfeb034b12a9a362109d78bec3/users
```

Response:

```
(200 OK):
[
    {
        "domain_id": "default",
        "email": "testuser@ibm.com",
        "enabled": true,
        "id": "106999c5eb7f4148932be6d24b6b772c",
        "name": "test-datauser-1",
        "options": {},
        "password_expires_at": null
    },
    {
        "domain_id": "default",
        "enabled": true,
        "id": "35470c08a699441394a0822da0161a16",
        "name": "sdadmin",
        "options": {},
        "password_expires_at": null
    },
    {
        "domain_id": "default",
        "email": "testuser@ibm.com",
        "enabled": true,
        "id": "dfdb2cadca0d4e9da26e6c40e5cadbd5",
        "name": "datauseruser",
        "options": {},
        "password_expires_at": null
    }
]
```

/auth/v1/collections/<collection_id>/role_assignments: GET

Returns a list of the users and groups and their roles that are assigned to the collection.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
√	X	[1]√	√	X
The specified collection must be one that is assigned to the requesting user ID. If no collection is assigned, then the command fails with error 403, "Not authorized".				

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/collections/<collection_id>/role_assignments
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get the list of groups to which the collection belongs:

Request:

```
curl -k -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/collections/a51f84bfeb034b12a9a362109d78bec3/role_assignments
```

Response:


```
{
    "groups": [
       {
            "description": "",
            "domain_id": "default",
            "id": "fccc7d34ee1d43128f2fda479f9f63e9",
            "name": "test_data_user_darko_group",
            "roles": [
                {
                    "domain_id": null,
                    "id": "73f471083b444117ba469d867c98c9b0",
                    "name": "datauser"
                }
            ]
        }
    ],
    "users": [
        {
            "description": "Test collection admin 3",
            "domain_id": "default",
            "email": "test_cadmin3@x.cmo",
            "enabled": true,
            "id": "154e1a1bb1fc40569924a0bd12ef7c0c",
            "name": "test_cadmin3",
            "options": {},
            "password_expires_at": null,
            "roles": [
                {
                    "domain_id": null,
                    "id": "55567000f3594155a1158edb6b5a9158",
                    "name": "collectionadmin"
                }
            ]
        }
}
```


Managing domains

You can create and manage user domains by using REST APIs.

- **/auth/v1/domains/<domain_ID>: GET**
  Gets the details of a specific domain.
- **/auth/v1/domains: POST**
  Creates a domain.
- **/auth/v1/domains/<domain_ID>: PATCH**
  Updates the attributes of an existing domain.
- **/auth/v1/domains/<domain_ID>: DELETE**
  Deletes a specific domain.

/auth/v1/domains/<domain_ID>: GET

Gets the details of a specific domain.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/domains/<domain ID>
```

Supported request types and response formats

Supported request types:

- GET

Supported response formats:

- JSON

Examples

The following example shows how to get details of a specific domain.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/domains/7a2251243611437cae707fb1aa5acf2c
```

Response:

```
(200 OK):
{
    "description": "Open LDAP Domain",
    "enabled": true,
    "id": "7a2251243611437cae707fb1aa5acf2c",
    "name": "ldap",
    "tags": []
}
```

/auth/v1/domains: POST

Creates a domain.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -X POST
https://<data_cataloging_host>/auth/v1/domains -d '<json domain details>'
```

In case of LDAP domain, you can specify the following attributes:

- "id"
- "type"
- "host"
- "port"
- "binddn"
- "bindpassword"
- "basedn"

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

The following example shows how to create the domain and pull in a list of LDAP users.

```
Type LDAP
Name LDAPForum
URL ( LDAP IP or host.domainame ) ldap.forumsys.com
Port 389
```

```
user cn=read-only-admin,dc=example,dc=com
password password
Suffix / Base DN dc=example,dc=com
Group Name Attribute cn
Group Object Class
Group Tree DN ou=mathematicians,dc=example,dc=com
Username Attribute uid
User Object Class person
User Tree DN dc=example,dc=com
```

/auth/v1/domains/<domain_ID>: PATCH

Updates the attributes of an existing domain.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/domains/<domain_ID> -X PATCH -d '<json_details_to_update>'
```

Supported request types and response formats

In an LDAP domain, you can update the following attributes:

- "id"
- "type"
- "host"
- "port"
- "binddn"
- "bindpassword"
- "basedn"

You can update one or more attributes in a single call.
Supported request types:

- PATCH

Supported response formats:

- JSON

Examples

The following example shows how to update the attributes of a domain:

Request:

```
curl -k -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
https://<data_cataloging_host>/auth/v1/domains/a51f84bfeb034b12a9a362109d78bec3 -X PATCH
-d '{"id":"7a2251243611437cae707fb1aa5acf2c", "type":"Default"}'
```

Response:

```
200 OK
```

/auth/v1/domains/<domain_ID>: DELETE

Deletes a specific domain.

The following table shows which roles can access this REST API endpoint:

Table 1. Access by role

Data admin	Data user	Collection Admin	Admin	Service user
X	X	X	✓	X

Synopsis of the request URL

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/domains/<domain_id> -X DELETE
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

The following example shows how to delete a specific domain.

Request:

```
curl -k  -H 'Authorization: Bearer <token>' https://<data_cataloging_host>/auth/v1/domains/
7a2251243611437cae707fb1aa5acf2c -X DELETE
```

Response:

```
204 No Content
```

Data protection

Data protection features in IBM Spectrum Discover support backup, disaster recovery, and restore operations to protect catalog metadata and system configurations.

- **Backup and restore**
  IBM Spectrum® Discover includes a set of procedures for safely backing up and restoring your database and file system for both Virtual appliance and the Red Hat® OpenShift® deployment.
- **Disaster recovery procedures**
  Disaster recovery procedures describe how to recover an IBM Spectrum Discover system from the loss of a single node or an entire deployment, including deployments on Red Hat OpenShift.

Backup and restore

IBM Spectrum® Discover includes a set of procedures for safely backing up and restoring your database and file system for both Virtual appliance and the Red Hat® OpenShift® deployment.

- **Backup and restore scripts**
  Backup and restore script is written in Python to run Db2 commands inside the Db2u pod in the OpenShift environment.

Backup and restore scripts

Backup and restore script is written in Python to run Db2 commands inside the Db2u pod in the OpenShift® environment.

Before you begin

1. Install the Python 3 and `oc` CLI and logged in to the environment.
2. Export the project name variable with the name of the environment. The default variable is `ibm-data-cataloging`.
3. For restore, check whether the `dcs-Backup` folder contains the tar.gz file that is generated from the previous backup.

About this task

The backup and restore script works in various steps, which require a stable IBM Data Cataloging environment. It must not be in use because it needs to scale IBM Data Cataloging pods to 0. The process consists of the following steps:

- Scale pods to 0
- Shutdown Db2
- Run backup or restore with Db2 move.
- Get the tar archive of backup files.
- Clean files from the Db2 pod
- Start Db2
- Scale pods to 1

Procedure

Follow the steps to run the script:

a. Get the script from the utility scripts repo resources repo then run it with the following command.

```
python3 backup_restore.py
```

b. The script contains a basic menu with the following options:
   i. Backup
   ii. Restore
   iii. Shutdown Db2

iv. Start Db2
v. Exit

The first option does not need any steps. It must start with the steps that are described in the overview. At the end, the backup tar.gz file must be in a directory that is named **dcs-Backup**, where the script runs.

The second option lists **dcs-Backup**tar.gz files. You can select one from those and run it with the steps that are described in the overview. After the restore is done, you need to go to the IBM Data Cataloging user interface and refresh the summary database to see the imported records.

After one of the first two options is finished, the system starts the scale pods. Some of them can fail in crashloop or error, but after the depending pods are running, they must recover after running state pods.

The third and fourth options are only used if the backup fails, leaving the environment in **maintenance mode**. To restore the state of IBM Data Cataloging, you need to scale pods manually.

Disaster recovery procedures

Disaster recovery procedures describe how to recover an IBM Spectrum Discover system from the loss of a single node or an entire deployment, including deployments on Red Hat OpenShift.

Use this process to recover from a disaster that involves an IBM Data Cataloging system and discusses the following scenarios:

- Recovery from the entire loss of a single node IBM Data Cataloging deployment.
- Recovery from the entire loss of an IBM Data Cataloging deployment on Red Hat® OpenShift®.

- **Preparations for disaster recovery**
  Before you need to perform disaster recovery, there are several tasks that must be accomplished to ensure the ability to recover.
- **Running disaster recovery for Red Hat OpenShift**
  The disaster recovery procedure for Red Hat OpenShift is similar to its backup and restore procedures.

Preparations for disaster recovery

Before you need to perform disaster recovery, there are several tasks that must be accomplished to ensure the ability to recover.

Procedure

1. Take a backup of the IBM Data Cataloging system as described in Backup and restore.
2. Record the installation configuration, including:
   - Network settings
   - Storage settings
   - CPU and memory
   - IBM Data Cataloging version
3. Ensure that the physical and virtual infrastructure is available to replace the system that might fail.
4. You cannot recover to a different version of IBM Data Cataloging. If a change of version is required, you need to recover to the current version before you install the upgrade.
   If you perform a recovery on a system that is upgraded, it is recovered directly to the upgraded IBM Data Cataloging version. For example, if IBM Data Cataloging 2.0.0.3 is deployed and then upgraded to IBM Data Cataloging 2.0.1 and then recovered from a failure, the recovery goes directly to the 2.0.1.

Running disaster recovery for Red Hat OpenShift

The disaster recovery procedure for Red Hat® OpenShift® is similar to its backup and restore procedures.

On successfully repairing or rebuilding your Red Hat OpenShift cluster, you can restore your system from an appropriate backup.

For more information, see Restoring deployed on.

.

Using a third-party data movement application to manage data

Introduction to data movement with IBM Data Cataloging.

Before you begin

- The third-party application must be registered with IBM Data Cataloging.
- IBM Data Cataloging must scan both the source and destination data sources before the data movement.
- The application must be configured to access the same source and destination data sources.

You can see the third-party application documentation for details.

About this task

The capabilities of IBM Data Cataloging can be extended by applications; for example, data mover applications can be used to move data between data sources, based on the IBM Data Cataloging insights. Applications register with IBM Data Cataloging providing details of the operations they support and the parameters they need to fulfill the operation.

The following table lists the supported data movement operations:

Table 1. Data movement operations

Operation	Source Type	Destination Type
MOVE	NFS	NFS
	S3	S3
	COS	COS
	SMB/CIFS	SMB/CIFS
COPY	NFS	NFS
	S3	S3
	COS	COS
	SMB/CIFS	SMB/CIFS
TIER	NFS	NFS
	S3	S3
	COS	COS
	SMB/CIFS	SMB/CIFS

Note:
For now, it is not possible for an application to register new operations beyond the records listed in the Data movement operations table.

You can create data movement policies that identify files and objects that are candidates for data operations, by using registered data movement Applications. The IBM Data Cataloging policy engine generates job request messages in JSON format. It contains a batch of files or objects with the source and target information, and any additional options such as preserving source system timestamps.

The job request messages are put onto a Kafka egress topic. The data movement application reads the messages from the topic and performs the necessary data movement operation, and provides response messages in JSON format on a Kafka ingress topic.

IBM Data Cataloging can interact with third-party data movement applications to move, copy, or tier data between data sources. Create data management policies on IBM Data Cataloging, and specify the set of documents to process by using a policy filter. The policy filter can be based on the system metadata or the custom metadata of documents that are collected by IBM Data Cataloging.

The third-party application registers with IBM Data Cataloging providing the operations that it supports (move, copy or tier). For each operation, it gives the list of parameter values that it needs to perform the operation.

When you create the data management policy in IBM Data Cataloging the user defines the filter, the parameter values, and when the policy must run. When the policy runs, IBM Data Cataloging sends the list of files to the data movement application and any additional parameters. The application processes the files and returns a status summary to IBM Data Cataloging. The summary is displayed to the user.

Note:

- During data migration, the migrated files need to preserve the IBM Data Cataloging tags. You can follow a manual procedure to preserve the tags. For more information, see Preserving tags during data movement.
- A third-party data movement application can register a schema, defining the parameters it requires for data movement. You need to set these parameters in the application user interface (UI) while creating the policy. Invalid parameters cause an error when you submit a policy.

  However, in few cases, due to the complexity of the schema, the error may not point to the precise location of the problem or may indicate that a valid parameter is invalid. Therefore, when an error is raised when you submit a policy, and it is not definite where the problem is, it is recommended that you check all the policy parameters.

Procedure

1. Log in to the IBM Data Cataloging GUI.
2. Go to Admin > Management Policies.
3. To create a policy, click Add Policy.
4. Click the slider control and set the status to one of the following values:

   Active
   : An active policy runs whenever its scheduling event is reached.

   Inactive
   : An inactive policy does not run even when its scheduling event is reached (including a NOW event).

5. Enter a policy name.
6. Enter a policy filter.
   The policy filter includes the criteria for selecting the files for moving or copying. For example, filetype="pdf" selects all files of type PDF.
   Note:
   - IBM Data Cataloging tracks the last known migration status for each file in the 'STATE' facet. This can be used when defining data movement policies to either target or avoid files in a particular state. The following values represent the file status:

migrtd (migrated)
: File contents are only present on the target system but a stub file exists on the source.

resdnt (resident)
: File contents are only present on the source system.

- The following are the scenarios where it is useful to filter the file status by 'STATE' facet are:
  - In a TIER policy where files that are already migrated shouldn't be migrated again.
  - In a COPY policy where files that are migrated shouldn't be copied as this would result in a recall of the data.
  - In a TIER policy where some files are being recalled in preparation for a workload. Any files that are already resident don't need to be recalled.
- To target files that are resident, add "state = 'resdnt'" to the filter criteria. To target files that are NOT resident, add "state <> 'resdnt'" to the filter criteria.

7. To select the policy type, click Next Step.
8. Select MOVE, COPY, or TIER as the policy type.
9. Select the agent name as the Agent.
10. Enter the remaining parameters. The parameters that are displayed depend on the application, and these parameters might include:

    Source connection type
    : Indicates the type of connection that the files currently reside on.

    Source connection
    : Indicates the name of the connection that the files currently reside on.

    Destination connection type
    : Indicates the type of connection that the files are being moved or copied to.

    Destination connection
    : Indicates the name of the connection name that the files are being moved or copied.

    Force migrate
    : Indicates whether to force demigration or recall of the file at source location when it is migrated to other location before you perform the operation.

    Overwrite
    : Indicates what value to give when a file exists at the destination.

    Preserve attributes, timestamp, or permissions
    : Indicates the parameters to control whether the files metadata is preserved.

11. To enter a schedule, select Next Step.
    The schedule indicates when you want to start the move of the copy.
12. To review the policy, select Next Step.
13. To create the policy, select Submit. The policy runs at the scheduled time.
14. When the policy runs or completes an execution status summary, view it by clicking Policy Preview.

- **Data movement with IBM Data Cataloging and Moonwalk**
  Policy-based data movement and management with integration of IBM Data Cataloging and Moonwalk applications.
- **Preserving tags during data movement**
  Preserve tags during data movement by generating a report of files and their tags before migration, then reapplying the tags using AUTOTAG policies after the destination connection is rescanned.

Data movement with IBM Data Cataloging and Moonwalk

Policy-based data movement and management with integration of IBM Data Cataloging and Moonwalk applications.

Moonwalk data mover application

Moonwalk is a heterogeneous Data Management System. It automates and manages the movement of data from primary storage locations to lower-cost file systems, object stores, tape, or cloud storage services. Files can be moved from source storage locations to target storage locations. These files are demigrated transparently when accessed by a user or an application. Moonwalk also provides capability to copy and move files along with a range of Disaster Recovery options.

Moonwalk provides a data movement application that integrates with IBM Data Cataloging, enabling to move, copy, or migrate/tier data from the IBM Data Cataloging user interface.

The moonwalk application is certified against the IBM Data Cataloging 2.0.4.

For more information about Moonwalk application and integrating the application with IBM Data Cataloging, see Moonwalk documentation.

Preserving tags during data movement

Preserve tags during data movement by generating a report of files and their tags before migration, then reapplying the tags using AUTOTAG policies after the destination connection is rescanned.

Before you begin

Generate a report of the files to be moved or copied before you start the data movement process. This report includes the relevant tags for each source file. A sample report is shown in the following table:

Table 1. Report generated for moving files

Path	Filename	Filetype	Datasource	tagA	tagB
/	Chrysanthemum.jpg	jpg	dir1	image	value1
/	Lighthouse.jpg	jpg	dir1	image	value1
/	newtextfile.txt	txt	dir1	text	value2
/	Hydrangeas.jpg	jpg	dir1	jpg	value1

Path	Filename	Filetype	Datasource	tagA	tagB
/	Thumbs.db	db	dir1	unknown	value1
/	Koala.jpg	jpg	dir1	jpg	value1
/	Desert.jpg	jpg	dir1	jpg	value1

About this task

If IBM Data Cataloging tags must be preserved during data migration, then you can follow a manual procedure to ensure that the tags are preserved in the moved or the copied files.

Procedure

1. Run the data movement process.
2. Rescan the destination connection after the data movement completes successfully.
3. Refer to the information provided in the report, shown in Table 1 that was generated before you started the data movement process, to identify the files for which you need to preserve the tags.
   Note: You can define a filter to identify and select the correct set of files that must be tagged. For example, if we want to move files shown in Table 1 to a new datasource (ddsource), you can define filters that include the actions to be taken for the tag values. For a filter defined as shown:

   ```
   datasource in ('ddsource') and filetype in ('jpg')
   ```

   You can create an AUTOTAG policy that sets the value of **tagA** to **image**.
   This process assumes either of the two things.
   - No other files belonging to these filetypes exist in the destination datasource.
   - Those files can also be tagged with the **tagA** values.
4. Reapply the tags (for example, **tagA**) on the moved or copied files by using the auto tagging capability of IBM Data Cataloging.
   Note:
   You must perform a search using the preceding filters to check that it retrieves the correct documents that are being selected. You can also sort, based on the report headings in MS Excel, to identify a suitable filter.

   It is not easy to identify a suitable filter to reapply the **tagB** values shown in Table 1 which indicates the difficulty in manually reapplying the tags.
   This solution is feasible only where you can define a small number of filters that identify the groups of files to which you need to apply the tags. If filters cannot be defined then you must run several AUTOTAG policies, which might be an impractical and tedious process.

Monitoring IBM Data Cataloging

This section provides information on how to monitor the IBM Data Cataloging capabilities.

- **Monitoring data sources**
  You can use the Home page to monitor the data sources that are connected to your IBM Spectrum® Discover environment. Use the Data Source Connections page view details about data source connections.
- **Monitoring the IBM Data Cataloging service environment**
  You can monitor the health and status of the IBM Data Cataloging service environment and obtain audit log information.
- **Health check monitoring**
  Provides series of checkpoints to check IBM Data Cataloging health status.

Monitoring data sources

You can use the Home page to monitor the data sources that are connected to your IBM Spectrum® Discover environment. Use the Data Source Connections page view details about data source connections.

- **Viewing data source status**
  Use the Home page to monitor your environment for storage system capacity, used capacity, records indexed, and duplicate files. You can also view data usage for a specific area of your organization.
- **Viewing data source connections**
  Use the Data Source Connections page to view connection information for the data sources that are connected to your IBM Data Cataloging environment.
- **Recommended to move**
  In the IBM Data Cataloging dashboard, you can categorize data as Recommended to move.
- **Deleting or editing a connection**
  Use the following information to delete or edit a connection.

Viewing data source status

Use the Home page to monitor your environment for storage system capacity, used capacity, records indexed, and duplicate files. You can also view data usage for a specific area of your organization.

The data in the home page is updated periodically. The last update is indicated by a time stamp.

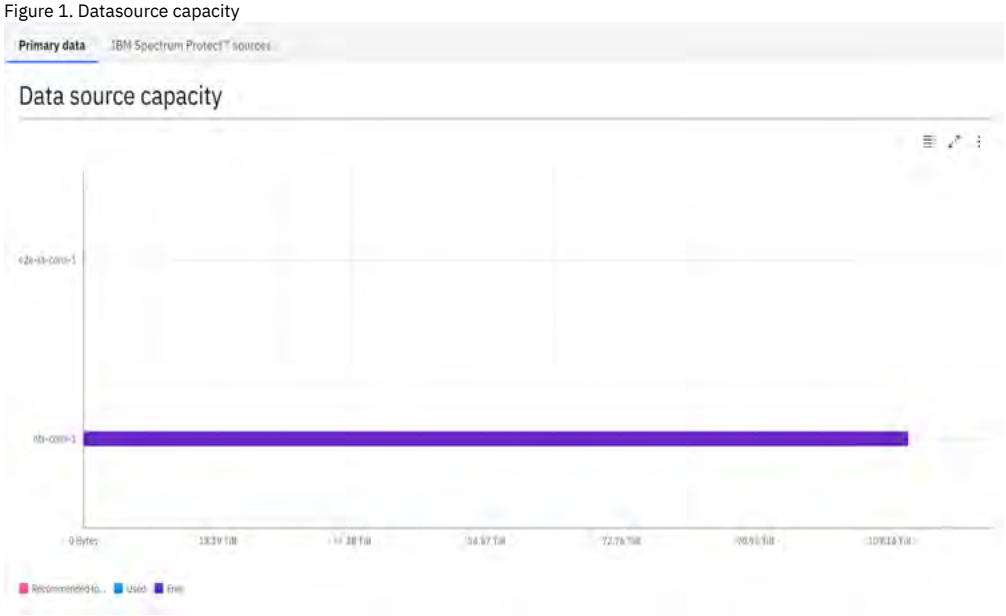
Viewing storage system capacity

Use the Data source capacity area to view capacity usage compared to the allocated capacity for all data sources that are registered with IBM Spectrum® Discover. The data sources can be a mixture of file systems and object vaults. A graph provides a convenient view of the current capacity of data sources and whether any are close to running out of space. This view also indicates the number of files to move or archive, based on user-defined policies.

Hover over a data source in the graph to view details about the data source. Click a data source to open the Search page and perform a search of the selected data source.

Note: Data sources that do not have data residing in them are not displayed in the graph.

Figure 1. Datasource capacity



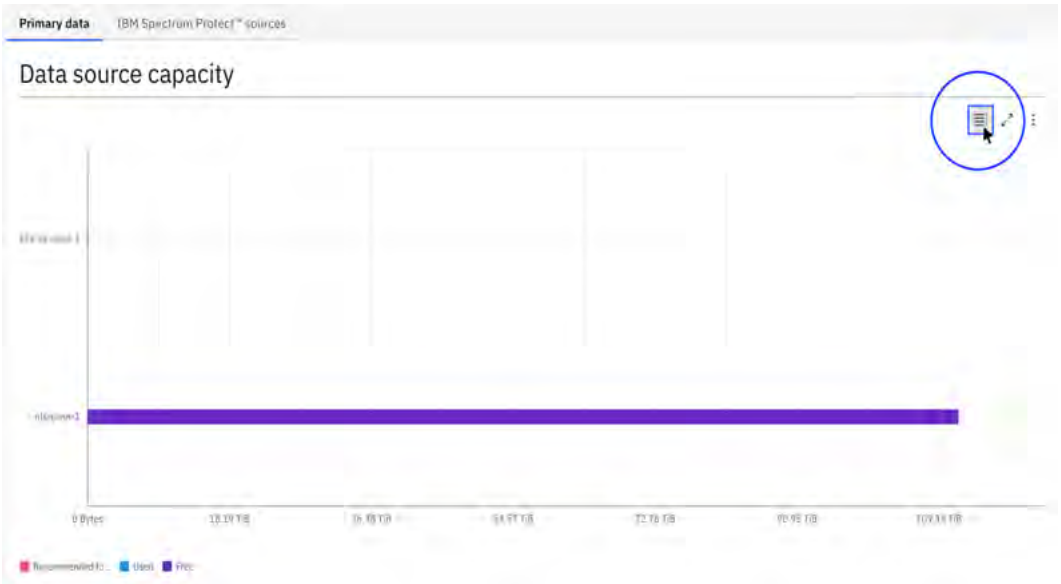
Understanding size and capacity differences

IBM Spectrum Discover collects size and capacity information. Generally:

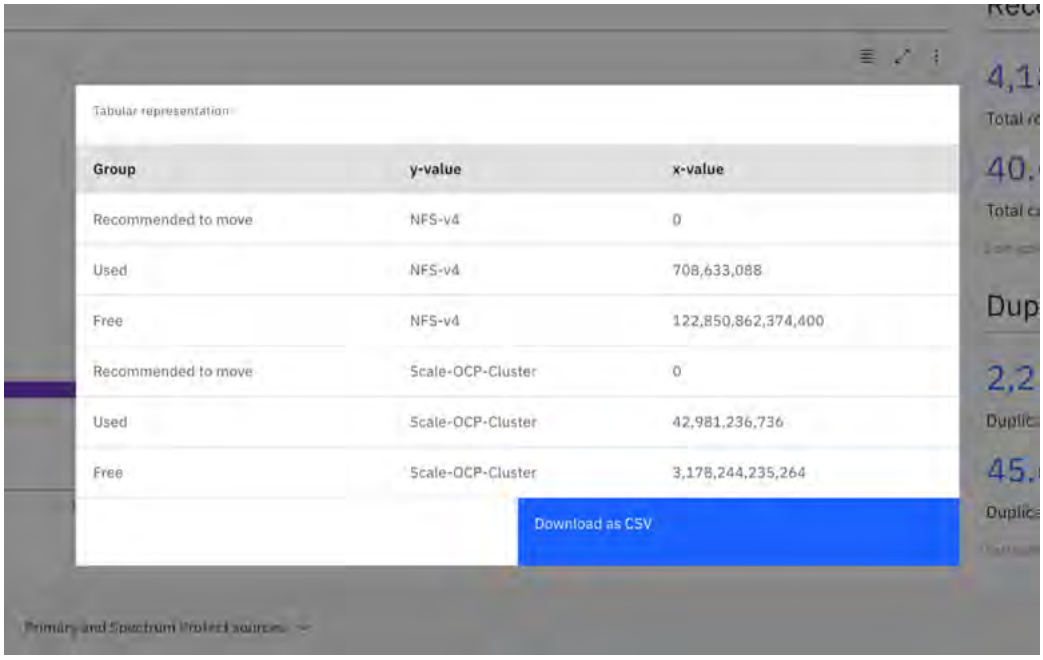
- Size refers to the size of a file or object in bytes.
- Capacity refers to the amount of space the file or object consumes on the source storage in bytes.

For objects, size and capacity values always match. For files, size and capacity values can be different because of file system block overhead or sparsely populated files.

Note: Storage protection overhead (such as RAID values or erasure coding) and replication overhead are not captured in the capacity values.
The Data source capacity area contains a view to visualize the data in a tabular representation. To see the view, click on the table button on the chart toolbar.



The tabular representation provides a complementary view to the analysis of the information on the chart. On the Group column, you can see the recommended to move, used, and free values that indicate the number of files to move or archive based on user-defined policies. On the y-value column, you can see the names of the data sources, and on the x-value column, you can see the size and capacity information in bytes.



Viewing used capacity

Use the Capacity Used by area to view graphs with an aggregated display of capacity usage for selected metadata attributes. You can view capacity for both primary and backup sources. The graphs provide details about capacity usage by aggregating across different attributes that are available from standard system metadata.

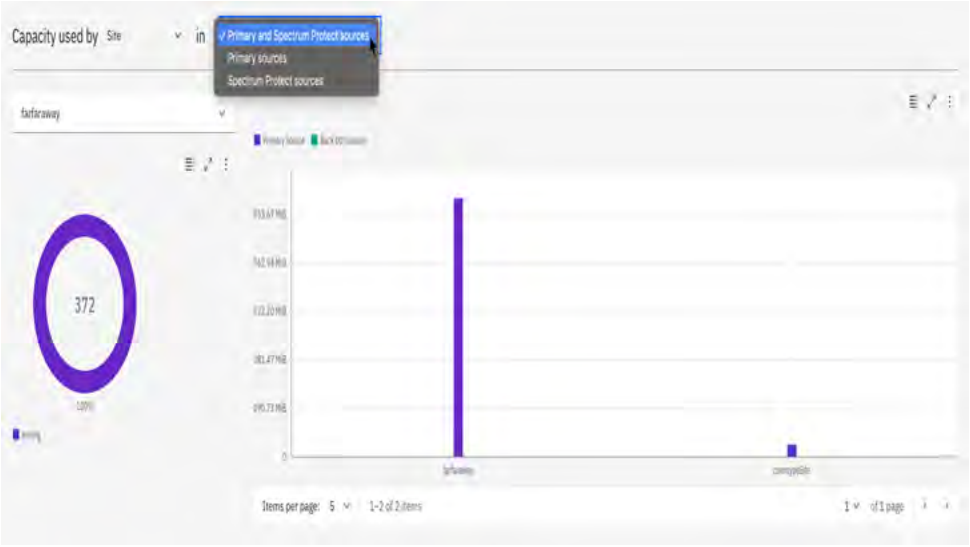
Use the Capacity Used by list to select an attribute and display the capacity consumers of that attribute in the graphs.

The Used graph displays the highest consumers of capacity for the selected attribute, in order of consumption.

The data source graph displays the percentage of overall usage per data source for the selected attribute. You can select a specific capacity consumer to display in the graph.

Hover over a value in a graph to view details. Click a value in a graph to open the Search page and search the selected item.

Figure 2. Example of the capacity that is being used



Viewing records indexed

Use the Records Indexed area to view both the total number of records and the capacity of the records that are indexed by IBM Spectrum Discover. This view provides a summary view of total storage usage.

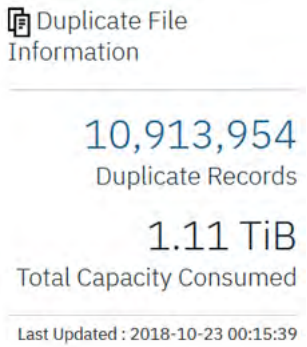


Click the Total Records Indexed value to open the Search page and perform a search of the indexed records.

Viewing duplicate file information

Use the Duplicate File Information area to view information about possible duplicate files within the storage environment. Possible duplicate files are files with the same name and size but different paths or object names. The number of duplicates and the capacity that is consumed by these files is displayed. You can also use a report that provides detailed and sorted information for the potential duplicates.

Click the Duplicate Records value to open the Search page and perform a search of duplicate records.

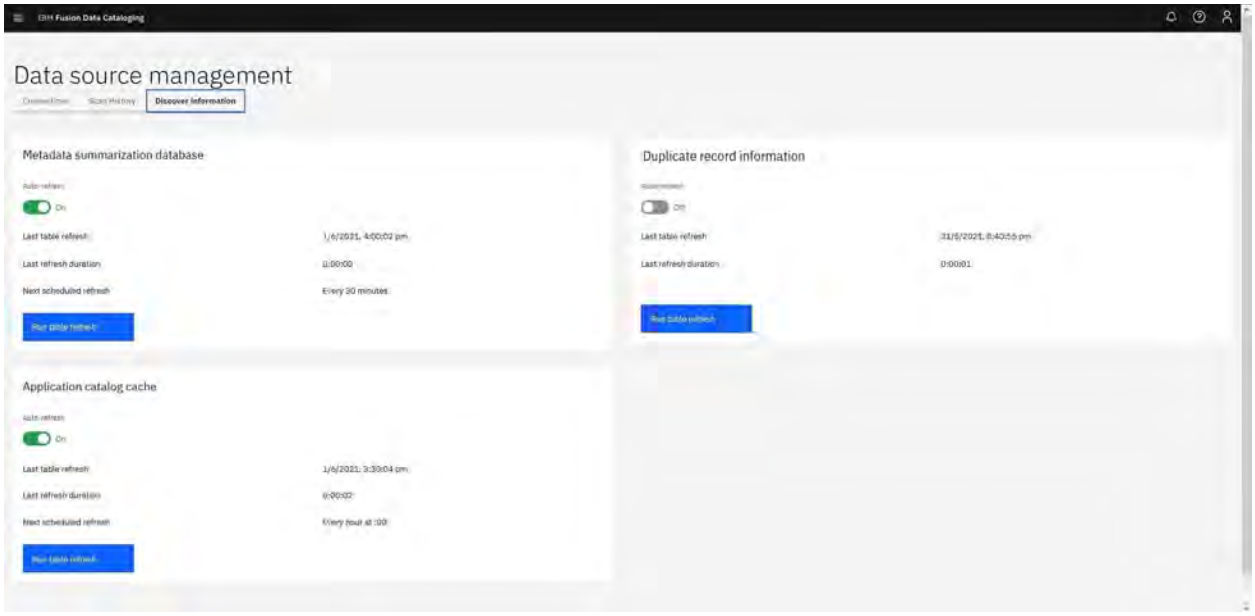


Identifying potential duplicates can be resource-intensive on IBM Data Cataloging. By default, the background task that refreshes potential duplicate information is disabled. However, you can update potential duplicate information either on demand or on a specific schedule. If you disable duplicate background task, the dashboard shows the following message:



To view and manage how often data in the home page is updated, navigate to Data connections > Discover database under Data source management window.

Figure 3. Run table refresh button in the Discover database window



From here, you can enable or disable the automatic updating of summary information. You can update information on the home page on demand by clicking the Run table refresh.

Viewing data source connections

Use the Data Source Connections page to view connection information for the data sources that are connected to your IBM Data Cataloging environment.

The following connections details are available:

Source Name
: A name that uniquely identifies the connection to the data source. A data source can have multiple connections.

Platform
: The platform of the data source - IBM Storage Scale system or IBM Cloud® Object Storage system.

Cluster
: The cluster address of the data source.

Data source name
: The full name of the data source.

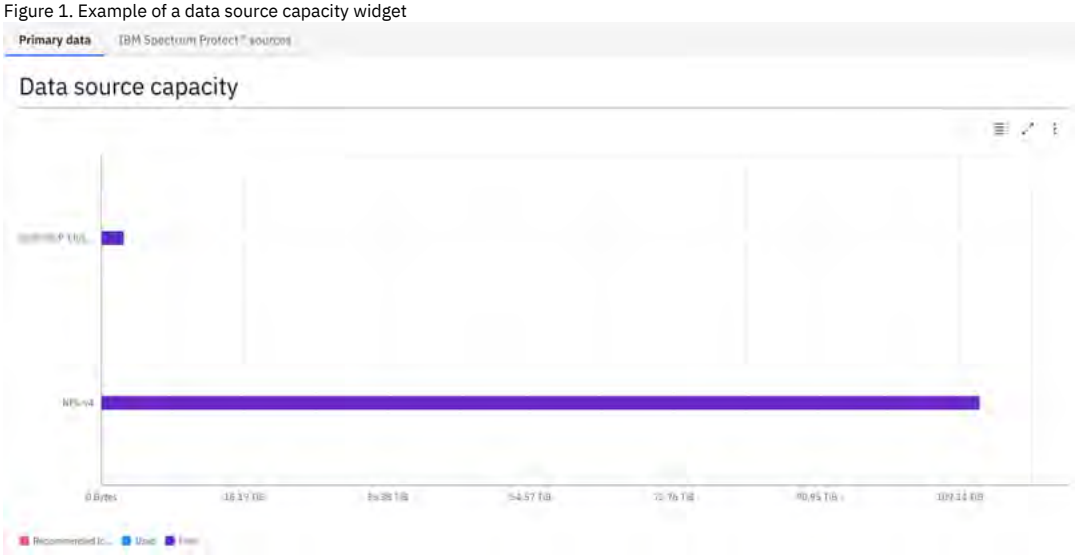
Site
: The physical location of the data source.

Recommended to move

In the IBM Data Cataloging dashboard, you can categorize data as Recommended to move.

Figure 1 shows an example of a data source capacity widget.

Figure 1. Example of a data source capacity widget



Use the **data source capacity** area to view capacity usage compared to the allocated capacity for all data sources that are registered with IBM Data Cataloging. The data sources can be a mixture of file systems and object vaults. A graph provides a convenient view of the current capacity of data sources and whether any are close to running out of space. This view also indicates the number of files to move or archive, based on user-defined policies.

Hover over a data source in the graph to view details about the data source. Click a data source to open the Search page and perform a search of the selected data source.

The data source capacity widget displays any files or objects that have the TEMPERATURE tag set to a value of ARCHIVE as Recommended to move. You can create an AUTOTAG policy to look for files and objects, which meet your archive criteria and set the TEMPERATURE tag to a value of ARCHIVE.

Any files that match the criteria of the AUTOTAG policy filter are tagged as ARCHIVE. The filter might be age-based or more complex. For example, the filter might match only certain file types, or files over some size threshold.

Figure 2 shows an example of a screen that shows the TEMPERATURE tag.

Figure 2. Example of a screen that shows the TEMPERATURE tag

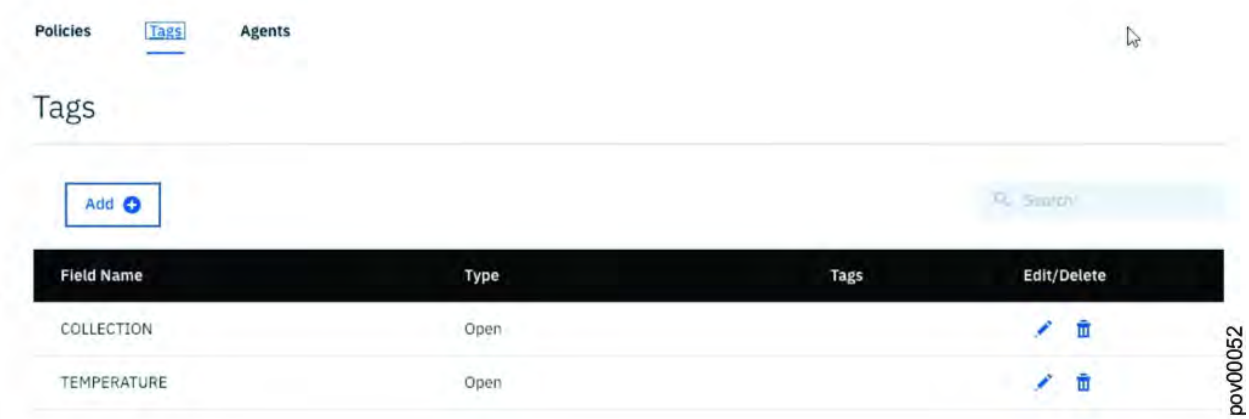


Figure 3 shows an example of an AUTOTAG policy to identify files and objects that have not been accessed for more than a year.

Figure 3. Example of an AUTOTAG policy to identify files and objects that have not been accessed in more than one year



Deleting or editing a connection

Use the following information to delete or edit a connection.

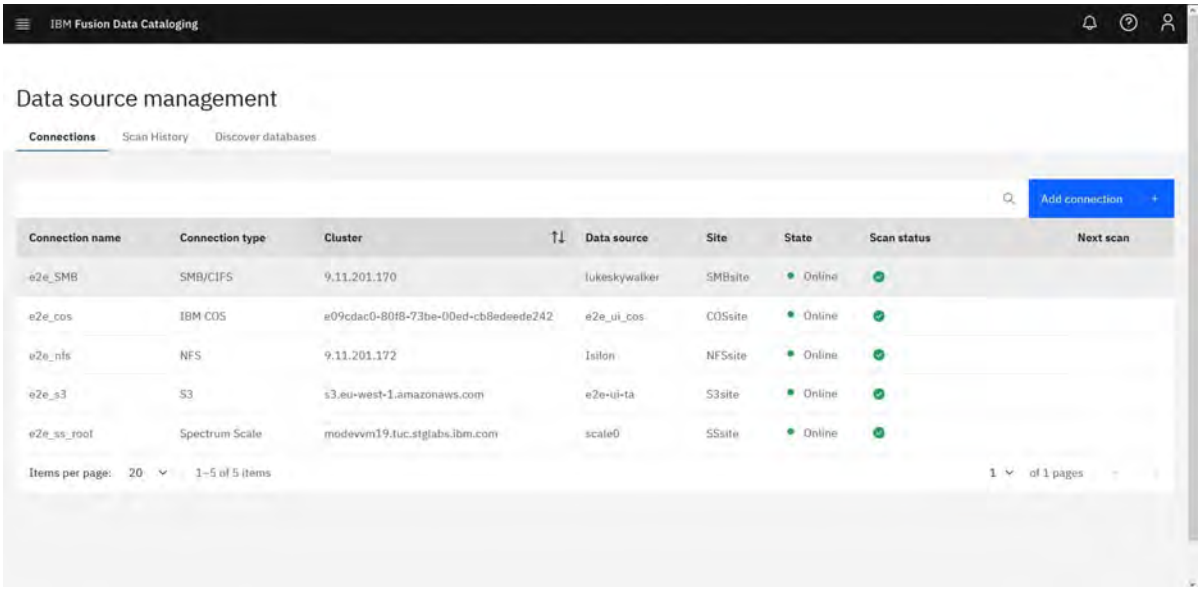
About this task

You can delete or edit a connection by using the graphical user interface.

Procedure

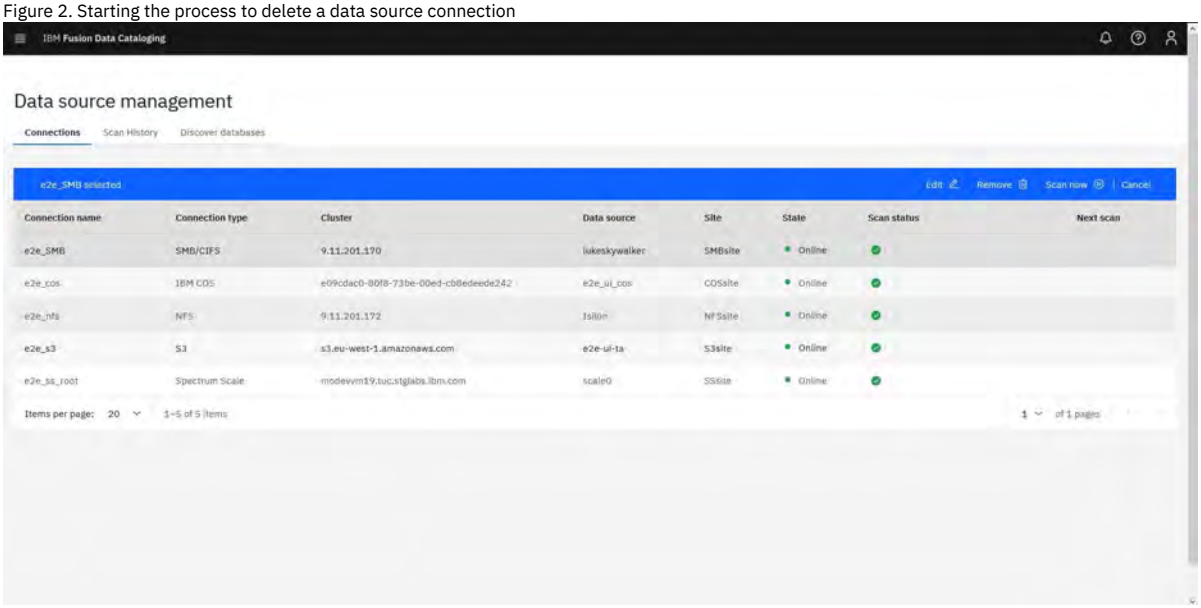
1. Click Data connections to display a listing of existing connections as shown in Figure 1

   Figure 1. Example of a listing of existing connections



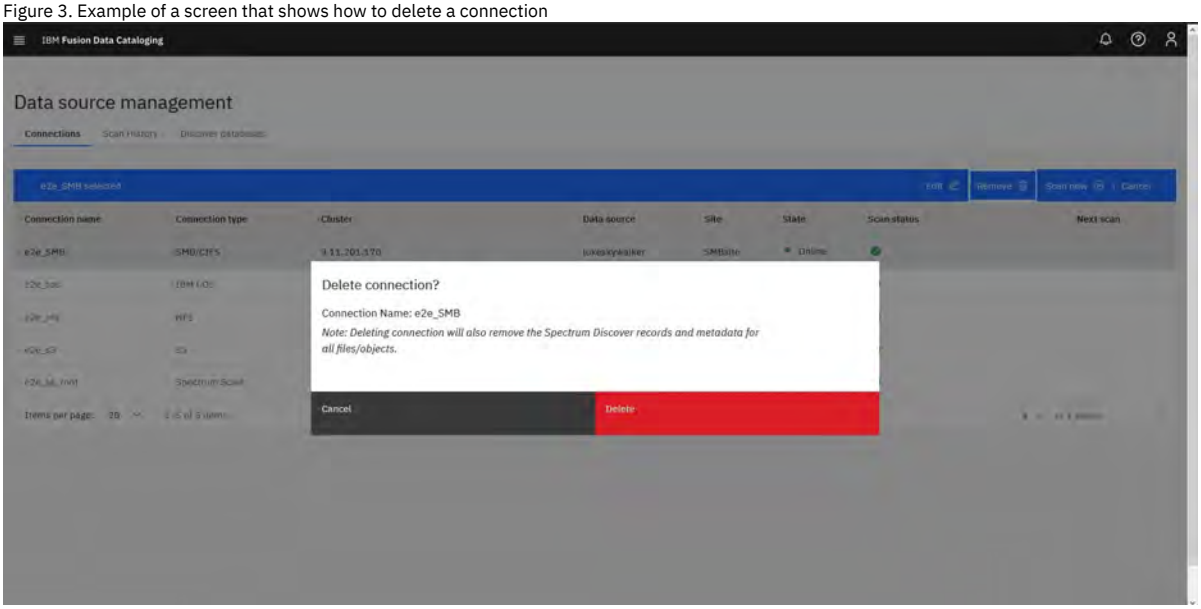
2. Click Remove to start the process to remove the data source connection as shown in Figure 2.

Figure 2. Starting the process to delete a data source connection



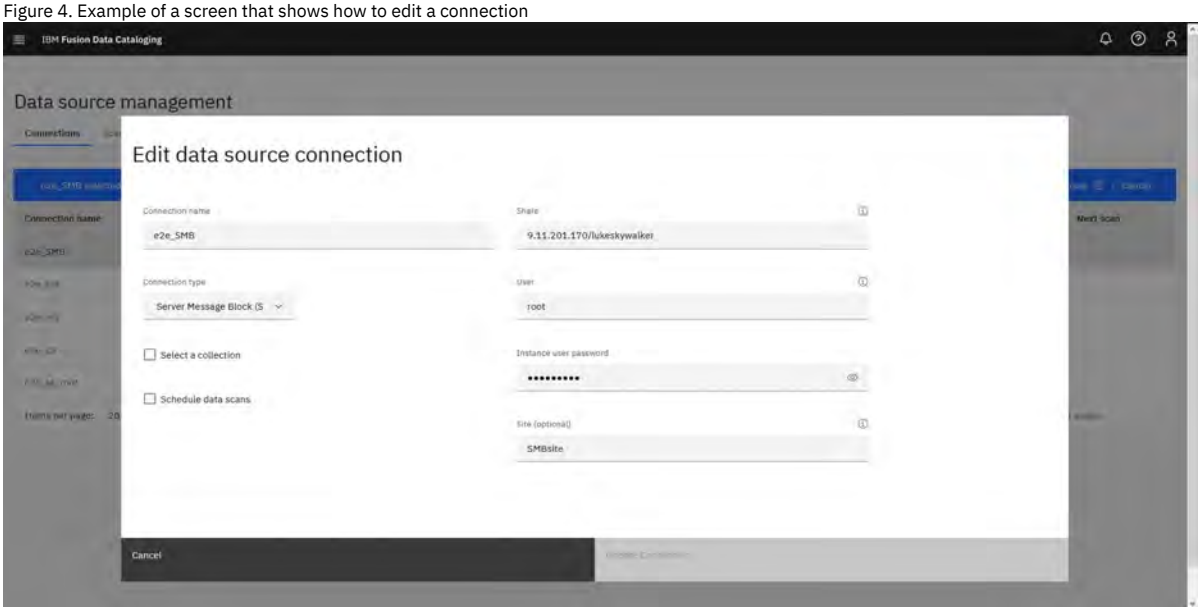
3. Clicking Remove displays a screen as shown in Figure 3. If you are sure you want to delete the connection, click Delete.

Figure 3. Example of a screen that shows how to delete a connection



4. To edit a connection, click Edit.
    a. Edit the appropriate fields in the window for Edit Data Source Connection.
    b. Click Update Connection.

Figure 4. Example of a screen that shows how to edit a connection



Monitoring the IBM Data Cataloging service environment

You can monitor the health and status of the IBM Data Cataloging service environment and obtain audit log information.

- **Audit log**
  Use the audit log entries to monitor activity of REST API calls within the IBM Spectrum® Discover environment, including the API endpoint that was used.
- **Configurable log trimmer handler**
  Log trimmer configurable feature provides the capability to interact with IBM Db2 components to remove unnecessary logs with a certain age, whose accumulation might impact storage and filesystem performance.
- **SD-Monitor**
  SD monitor obtains metrics about the Kafka messages that are fetched from the Kafka topics.

Audit log

Use the audit log entries to monitor activity of REST API calls within the IBM Spectrum® Discover environment, including the API endpoint that was used.

To view audit log entries, extract the output from the compressed file that is generated by the FFDC script. You can use a text editor to read the FFDC output. Audit log entries are in JSON format and are identified in the FFDC output by the string AUDIT in the type field.

For more information about API endpoints in the IBM Spectrum Discover environment, see *REST API* in *IBM Data Cataloging: REST API Guide*REST API for IBM Data Cataloging.

The audit log includes the following fields:

service
: The service that processed the request. The service and node name are included. The following details are optional: **`namespace`**, **`serviceInstance`**, and **`containerId`**.

requestId
: The request ID that is returned back to the client, or a correlation tag that is used for internal tracking.

timestampStart
: The time that the request was received.

request
: The API endpoint that made the request.

serverAddress
: The IP address of the server or node that processed the request.

userAgent
: The identification string of the user agent that made the request.

type
: The log entry type: **`AUDIT`**

responseSize
: Size of the response, in bytes, sent back to the client.

hostname
: The IP address from which the request originates.

protocol
: The protocol of the request.

requestLatency
: The latency of the request in milliseconds.

responseStatus
: The return code that is provided to the client.

auth
: The user name and the authentication scheme, bearer (for LDAP) or basic (for local authentication).

Configurable log trimmer handler

Log trimmer configurable feature provides the capability to interact with IBM Db2 components to remove unnecessary logs with a certain age, whose accumulation might impact storage and filesystem performance.

By default, the job to erase log files is configured to run at midnight daily and erase logs older than 12 hours.

To configure age for log clean up, a config map is defined. Use the following example to configure it:

```
export NAMESPACE=ibm-data-cataloging
export TIME_DELETION=720
oc patch configmap db2-log-trimmer -n $NAMESPACE --type merge -p "{\"data\":{\"time-limit-min\":\"${TIME_DELETION}\"}}"
```

TIME_DELETION vary in minutes.

SD-Monitor

SD monitor obtains metrics about the Kafka messages that are fetched from the Kafka topics.

About this task

The SD monitor provides information about the Kafka messages that are fetched from the Kafka topics.

In the IBM Data Cataloging service, each Kafka topic represents a type of connection, such as NFS, IBM Storage Scale, COS, and S3, which has multiple consumers fetch the messages that are sent from the producers when they connect to a data source and scan data from it.

You can retrieve the following metrics from the SD monitor:

- Current committed offset for the set of topics and partitions.
- Log end offset (highest offset in the partition).
- Lag, the difference between the current committed offset and log end offset values.

Procedure

Follow the steps to get SD monitor information is as follows:

a. Run the following command to get the logs of the SD monitor pod.

```
(oc logs $( oc get pods -n ibm-data-cataloging | grep sdmonitor | awk '{print $1}' ) -n ibm-data-cataloging) | grep Fetch
```

b. Locate the array that contains the metrics about the connection type that you want to debug.
An example array for NFS connections.

```
'file-scan-topic': {0: [30768, 30768, 0], 1: [31050, 31050, 0], 2: [30897, 30897, 0], 3: [30788, 30788, 0], 4 : [30608,
30608, 0], 5: [30525, 30525, 0], 6: [30908, 30908, 0], 7: [30610, 30610, 0], 8: [30726, 30726, 0], 9 : [30959, 30959, 0]}
```

Note: Each sub-array `0: [30768, 30768, 0]` indicates the metrics that are related to one of the consumers. In this sub-array, the first value represents the current committed offset. The second value is the log end offset, and the third value represents the lag.
It is important to consider the following points when you read the log information:

- The SD monitor provides a condensed metric of the Kafka messages that are received by each one of the consumers.
  For example, if you run two different NFS scans, the expectation is to get the metrics of all the Kafka messages sent by the two scans.
- The SD monitor runs every 30 seconds, which means that you can see a small lag between the progress of the scans and the metrics they are reported in the SD monitor log.

Health check monitoring

Provides series of checkpoints to check IBM Data Cataloging health status.

It provides IBM Data Cataloging health status and different commands that might be used for continuous monitoring of the health check.

When scanning is running, different checkpoints can be made to ensure that they are scanned properly.

Monitor connection manager and db2whrest pods frequently to avoid overhead on database
: One of the issues that might disrupt IBM Data Cataloging function is overrunning the database, causing malfunctions in different components such as the user interface or connection manager.
The top memory and CPU consumption for connection manager and db2whrest pods is around 85% of their limits.
It is recommended to run the following commands to ensure that those limits are met for regular usage:

1. Check the current usage of connmgr pod and db2whrest.

```
$oc adm top pod isd-connmgr-795544f666-l25mc
$oc adm top pod isd-db2whrest-7c574c7bf-x719t
```

Example output:

```
$oc adm top pod isd-connmgr-795544f666-l25mc
NAME                                CPU(cores)   MEMORY(bytes)
isd-connmgr-scheduler-795544f666-l25mc   0m           34Mi

$oc adm top pod isd-db2whrest-7c574c7bf-x719t
NAME                             CPU(cores)   MEMORY(bytes)
isd-db2whrest-7c574c7bf-x719t   21m          512Mi
```

2. Compare current usage with the limits set and make sure that it is not greater than 85% for a long period of time.

```
$oc describe pod isd-connmgr-795544f666-l25mc|grep -A2 Limits
    Limits:
      cpu:     1
      memory:  4Gi
$oc describe pod isd-db2whrest-7c574c7bf-x719t|grep -A2 Limits
    Limits:
      cpu:     2
      memory:  8Gi
```

Note: IBM Data Cataloging recommends having 10 parallel scans at most to avoid malfunction or reduce significantly performance on the product.

Check request latency of a connection filtering db2whrest log

This checkpoint is important for checking the time for a request query to be processed, the following example uses one connection to see the time for the request to respond. Values < 2000 (milliseconds) are expected.

1. This checkpoint is important for checking the time for a request query to be processed, the following example uses one connection to see the time for the request to respond. Values < 2000 (miliseconds) are expected.
2. Run the following command to check the pod that is associated with db2whrest.

```
$oc get pods | grep db2whrest
```

Example output:

```
$oc get pods | grep db2whrest
isd-db2whrest-7x574c7bf-c7l9t           1/1     Running         0       43h
```

3. Get the logs and filter them.
For example:

```
$oc logs isd-db2whrest-7x574c7bf-c7l9t|grep -E '${connectionName}.*requestLatency'
{"type": "AUDIT", "hostname": "172.17.47.184", "serverAddress": "172.21.166.189", "userAgent": "python-requests/2.31.0",
```

4. It is also acceptable if the requested latency value is less than 2000. This is an important checkpoint that can give you clues about database behavior when you submit a new scan.

Monitor Kafka lag as complementary signal of a healthy product

When a scan is being processed, one signal of a healthy environment is the lag, which is the difference between indexed records and scanned records.
A healthy state would be when the lag is less than 30%. A state of unhealthiness occurs when the number of scanned records continues to increase without progress in indexing, resulting in a significant lag and a sustained difference more than 30% for an extended period.
To resolve that situation, it is recommended to reduce the number of parallel scans, giving more time for the ingestion rate to increase properly and reducing the lag.

Check average insert rate of a scan and batch size

The following checkpoints give clues to the health of the insert rate into the database and the batch size of Kafka.

1. Insert rate values might vary depending on different factors. However, the expected rate is around 200 messages per second. It means 200 messages per second get inserted into the database per consumer.
2. Regarding the batch size of Kafka, it is the amount of data that is accumulated and sent as a batch by a Kafka producer. This value is expected to be around 50,000 overall. If the value decreases significantly for a long period, it might be a sign of misbehavior.
3. Run the following command to get the consumer pod details.

```
$oc get pods|grep consumer-file-scan
```

Note: In this case, the consumer is the file, because it is an NFS scan and should be 10 pods.
Example output:

```
isd-consumer-file-scan-6d6bbf8487-45c6c                           1/1     Running     0               7d13h
isd-consumer-file-scan-6d6bbf8487-6zfv9                           1/1     Running     0               7d13h
isd-consumer-file-scan-6d6bbf8487-77g7v                           1/1     Running     0               7d13h
isd-consumer-file-scan-6d6bbf8487-97cfv                           1/1     Running     0               7d13h
isd-consumer-file-scan-6d6bbf8487-bwmr5                           1/1     Running     0               7d13h
isd-consumer-file-scan-6d6bbf8487-c5cb9                           1/1     Running     0               7d13h
isd-consumer-file-scan-6d6bbf8487-ljkws                           1/1     Running     0               7d13h
isd-consumer-file-scan-6d6bbf8487-m96hc                           1/1     Running     0               7d13h
isd-consumer-file-scan-6d6bbf8487-w55lr                           1/1     Running     0               7d13h
isd-consumer-file-scan-6d6bbf8487-z8hqd                           1/1     Running     0               7d13h
```

4. Get the logs and filter them.
For example:

```
$oc logs isd-consumer-file-scan-6d6bbf8487-45c6c|grep -A5 "Avg Insert"
… omitted …
DB Avg Insert Rate (msg/sec): { current: 210, min: 1, max: 245 }
2023-06-22 13:54:40.611 > offset_commit_cb: success, offsets:[{part: 1, offset: 79452963, err: none}]
2023-06-22 13:56:25.588 > Consumed= 50014 Dup_id= 13 Skipped= 0 |Last_Batch: Att= 50000 Succ= 50000 Fail= 0 |Total:
```

```
Att= 50001 Succ= 50001 Fail= 0 | Batch_Time(ms): E= 160 B= 36155 L= 68659 KC= 0 TH= 0 Tot= 104974
 DB Avg Insert Rate (msg/sec): { current: 192, min: 1, max: 245 }
2023-06-22 13:56:25.605 > offset_commit_cb: success, offsets:[{part: 1, offset: 79502976, err: none}]
2023-06-22 13:59:44.298 > Consumed= 100024 Dup_id= 23 Skipped= 0 |Last_Batch: Att= 50000 Succ= 50000 Fail= 0 |Total:
Att= 100001 Succ= 100001 Fail= 0 | Batch_Time(ms): E= 139 B= 11888 L= 186663 KC= 1 TH= 0 Tot= 198691
 DB Avg Insert Rate (msg/sec): { current: 196, min: 1, max: 245 }
2023-06-22 13:59:44.314 > offset_commit_cb: success, offsets:[{part: 1, offset: 79552986, err: none}]
2023-06-22 14:03:04.724 > Consumed= 150031 Dup_id= 30 Skipped= 0 |Last_Batch: Att= 50000 Succ= 50000 Fail= 0 |Total:
Att= 150001 Succ= 150001 Fail= 0 | Batch_Time(ms): E= 155 B= 1594 L= 197868 KC= 1 TH= 0 Tot= 200408
 DB Avg Insert Rate (msg/sec): { current: 206, min: 1, max: 245 }
```

Check Qpart assignation when a new connection is submitted for scan

This checkpoint shows that when a scan is assigned to the different partitions, the health status must be associated with the correct assignment of the 10 Qparts (0–9), and scangen must be constant.

1. Run the following command to get the producer pod details.

   ```
   $oc get pods|grep producer-file-scan
   ```

   Note: In this case, the producer is the file, because it is an NFS scan only one producer is acceptable.
   Example output:

   ```
   $oc get pods|grep producer-file-scan
   isd-producer-file-scan-79c45bbc77-jm5v6                           1/1     Running     4 (7d14h ago)   7d14h
   ```

2. Get the logs from one and filter the connection name and **COMMITSCAN**.
   For example:

   ```
   $oc logs isd-producer-file-scan-79c45bbc77-jm5v6|grep -E ${connectionName}.*COMMITSCAN'
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:0]
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:1]
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:2]
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:3]
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:4]
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:5]
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:6]
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:7]
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:8]
   Built commit scan message [connection:${connectionName},eventname:COMMITSCAN,scangen:1,qpart:9]
   ```

   In case 1, it is acceptable for the connection to be assigned to 10 different partitions and scangen to be constant.

Check connections progressing and scan in progress state when a new scan started

These requests can be made to the connection manager API endpoint to check the status of a connection. It would be necessary to create a token to be ready to consume the API properly. For more information about how to get the token, see /auth/v1/token: GET.
At this stage, the token is saved in the **${TOKEN}** variable, **hostRoute** and **connectionName** are identified, and **jq** is installed for JSON formatting purposes.

For example:

```
$ curl -q -H "Authorization: Bearer $TOKEN" https://<<hostRoute>/connmgr/v1/connections/<<connectionName>>|jq
{
  "name": "exports",
  "platform": "NFS",
  "cluster": "nfs-1.nfs.svc.cluster.local",
  "datasource": "exports",
  "current_gen": 3,
  "password": null,
  "site": "",
  "online": 1,
  "scan_topic": "file-scan-connector-topic",
  "le_topic": "file-le-connector-topic",
  "le_enabled": 1,
  "host": "nfs-1.nfs.svc.cluster.local",
  "mount_point": "/exports",
  "protocol": "nfs",
  "user": null,
  "additional_info": "{\"working_dir\": \"/nfs-scanner/\", \"local_mount\": \"/nfs-
scanner/da68f4b426134c5d9fb498a15219857f\"}",
  "scan_in_progress": 1,
  "total_records": 10162,
  "scan_compl_rec": 0,
  "schedule": null
}
```

As shown, there is a scan in progress for that specific connection, and total records must be augmenting over time. It would be necessary to give some seconds to get data reflected on the endpoint.
These are some of the checkpoints that can be used for checking the state of the connections. This was one example of the many endpoints that can be consumed by the IBM Data Cataloging API. For more information, see REST API for IBM Data Cataloging.

Check database records increasing while scan is progressing

If querying the API or scanning progress on the user interface does not show any progress, it might give the impression that the system is not responding properly. However, that might not be accurate. It is necessary to establish a connection to the database and make some queries to ensure that the system is actually making insertions into the proper table.
Warning: The database is stressed when doing several transactions. Be careful when you query the database directly. Also, avoid doing complex queries when one or several scans are occurring; this might cause unnecessary overhead and impact the unusual behavior of the database.

1. Run the following command to get db2 head pod and access it.

   ```
   $ headPodName=$(oc -n ${projectName} get po --selector name=dashmpp-head-0|grep isd|awk '{print $1}')
   $ oc -n ${projectName} rsh ${headPodName}
   ```

2. Change user to db2inst1 and connect to bludb database.

```
sh-4.4$ su db2inst1 -
[db2inst1@c-isd-db2u-0 - Db2U ]$ db2 connect to bludb

   Database Connection Information

 Database server        = DB2/LINUXX8664 11.5.7.0
 SQL authorization ID   = DB2INST1
 Local database alias   = BLUDB

#Do couple counting queries agains metaocean table and make sure it's progressing leaving couple of seconds from one
query to another

[db2inst1@c-isd-db2u-0 - Db2U ]$ select count(fkey) from bluadmin.metaocean

1
--------------------------------
                           10364.

  1 record(s) selected.

[db2inst1@c-isd-db2u-0 - Db2U ]$ select count(fkey) from bluadmin.metaocean

1
--------------------------------
                           10952.

  1 record(s) selected.
```

Collecting logs and metrics

Steps to collect logs and metrics for the IBM Data Cataloging.

Procedure

1. Run the following `oc` login command.

   ```
   oc login --token=<OCP_TOKEN> --server=<OCP_SERVER>
   ```

   For more information about CLI login, see CLI login.

2. Export `OCP_TOKEN`, `PROJECT_NAME`, and `TIMESTAMP` environment variables.

   ```
   export OCP_TOKEN=$(oc whoami --show-token) # or the actual token if not logged in
   export PROJECT_NAME=ibm-data-cataloging
   export TIMESTAMP=$(date +"%Y%m%d%H%M%S")
   ```

3. Run the following command to create a secret with a token to use in the script that collects resources.

   ```
   oc -n $PROJECT_NAME create secret generic dcs-must-gather --from-literal=token=$OCP_TOKEN
   ```

4. Create dcs-must-gather-pod.yaml file.

   ```
   cat >> dcs-must-gather-pod.yaml << EOF

   apiVersion: v1
   kind: Pod
   metadata:
     name: dcs-must-gather
     labels:
       app: isd
       component: discover
       role: must-gather
   spec:
     containers:
     - name: must-gather
       env:
         - name: WORKDIR
           value: /tmp
         - name: OCP_TOKEN
           valueFrom:
             secretKeyRef:
               name: dcs-must-gather
               key: token
       image: cp.icr.io/cp/ibm-spectrum-discover/must-gather:2.1.3
       imagePullPolicy: Always
       resources:
         requests:
           cpu: 50m
           memory: 128Mi
           ephemeral-storage: 2Gi
         limits:
           cpu: 2
           memory: 8Gi
           ephemeral-storage: 5Gi
       command:
         - '/bin/bash'
         - '-c'
         - '/opt/must-gather/must-gather.sh; while true; do sleep 60; done'

   EOF
   ```

5. Run the following command to create a must gather pod.

   ```
   oc -n $PROJECT_NAME create -f dcs-must-gather-pod.yaml
   ```

6. Monitor the logs until it reports that the `tar` file has been created.

   ```
   oc -n $PROJECT_NAME logs -f dcs-must-gather
   ```

   Example output:

   ```
   2023-09-12 04:12:12 - INFO - Archiving completed: dcs-must-gather.tar.xz
   ```

7. Copy the `tar` file.

   ```
   oc -n $PROJECT_NAME cp dcs-must-gather:/tmp/dcs-must-gather.tar.xz dcs-must-gather-$TIMESTAMP.tar.xz
   ```

8. Remove the must gather resources.

   ```
   oc -n $PROJECT_NAME delete pod dcs-must-gather
   oc -n $PROJECT_NAME delete secret dcs-must-gather
   ```

Data Cataloging Harvester CLI

IBM Data Cataloging Harvester is a new capability that is designed to import external data to the IBM Data Cataloging service catalog database. It imports the data even if it is not coming from a Db2 database that uses the same schema.

Before you begin

- Ensure you have the following installations:
  - Python +3.11 or later version
  - `curl tar gzip sed pip3` packages
  - Red Hat® OpenShift® CLI.

With the IBM Data Cataloging Harvester, it is possible to import the external data to the catalog database even if it is not coming from a Db2 table using the exact same schema. IBM Fusion 2.8.0 supports importing data associated to a single data source also known as a connection. After using the CLI against a valid SQLite file, then it should create a connection and import all metadata stored in the import file records or additional tags depending on the command that is executed.

Supported import formats

The Harverster CLI helps to read records in the specified format and sends them to the import service component that runs on the cluster to use by Db2 after it is validated. Not only the input file is in the format that the Harvester supports, but also the schema rules that succeed during analyzing tasks.

The IBM Data Cataloging harverster needs the following schemas. The IBM Data Cataloging harvester requires at least two tables to work with:

Connections table
: It is used to store information about the data source.

```
CREATE TABLE CONNECTIONS (
    name TEXT,
    platform TEXT,
    cluster TEXT,
    datasource TEXT,
    site TEXT,
    host TEXT,
    mount_point TEXT,
    application_code TEXT,
    local_mount TEXT,
    total_records INTEGER,
    current_gen TEXT,
    password TEXT,
    online TEXT,
    scan_topic TEXT,
    le_topic TEXT,
    le_enabled INTEGER
)
```

Metaocean table
: It contains all the base metadata present in the data source.

```
CREATE TABLE METAOCEAN (
    INODE INTEGER,
    OWNER TEXT,
    [GROUP] TEXT,
    PERMISSIONS TEXT,
    UID INTEGER,
    GID INTEGER,
    PATH TEXT,
    FILENAME TEXT,
    MTIME TEXT,
    ATIME TEXT,
    CTIME TEXT,
    SIZE INTEGER,
    TYPE TEXT,
    STATE TEXT
)
```

The SQLite database might have more optional tables if you want to import tag values for additional metadata. The additional tables cannot be processed unless the tags mode in the Harvester CLI defines tables by their names.

Example of additional tags table:
Note: Inode and filename are only the required columns, and rest are the names of already created tags on Data Cataloging.

```
CREATE TABLE META (
    inode TEXT,
    filename TEXT,
    fkey TEXT,
    ExampleCustomTag1 TEXT,
    ExampleCustomTag2 TEXT,
    ExampleCustomTag3 TEXT,
)
```

Setting up the external host

1. Log in to the Red HatOpenShift CLI. For procedure, see Red Hat OpenShift CLI.
2. If the Harvester CLI runs on hosts outside of the OpenShift cluster, then it is possible to face SSL problems caused by self-signed certificates. To resolve SSL problems, you can use the following commands to download the router certificate in the host that running the Harvester and then use environment variables to use them.
    a. Run the following command to download the OpenShift router TLS certificate.

    ```
    oc -n openshift-ingress get secret router-certs-default -o json | jq -r '.data."tls.crt"' | base64 -d > $(pwd)/router_tls.crt
    ```

    b. Run the following command to export the environment variables to define the path of the TLS certificate.

    ```
    export IMPORT_SERVICE_CERT=$(pwd)/router_tls.crt
    export DCS_CERT=$(pwd)/router_tls.crt
    ```

3. Download the Harvester CLI.

    ```
    export DCS_NAMESPACE=ibm-data-cataloging
    export DCS_IMPORT_SVC_HOST=$(oc -n $DCS_NAMESPACE get route isd-import-service -o jsonpath="{.spec.host}")
    export DCS_CONSOLE_HOST=$(oc -n $DCS_NAMESPACE get route console -o jsonpath="{.spec.host}")
    export DCS_USERNAME=$(oc -n $DCS_NAMESPACE get secret keystone -o jsonpath="{.data.user}" | base64 -d)
    export DCS_PASSWORD=$(oc -n $DCS_NAMESPACE get secret keystone -o jsonpath="{.data.password}" | base64 -d)

    curl "https://${DCS_IMPORT_SVC_HOST}/v1/download_harvester" --output harvester.tar.gz --insecure
    tar xf harvester.tar.gz
    rm -vf harvester.tar.gz
    ```

4. Run the following set up script to start by using the Harvester CLI.

    ```
    chmod +x ./harvester/setup.sh
    ./harvester/setup.sh
    ```

Running the Harvester CLI

1. Ensure that you export the records with all the base metadata.

    ```
    SQLITE_FILE_PATH=/tmp/example.sqlite
    CONFIG_FILE=$(pwd)/harvester/config.ini
    python3 harvester metaocean $SQLITE_FILE_PATH -c $CONFIG_FILE
    ```

2. Use `-p nfs` option if you want to import NFS-based metadata.

    ```
    SQLITE_FILE_PATH=/tmp/example.sqlite
    CONFIG_FILE=$(pwd)/harvester/config.ini
    python3 harvester metaocean $SQLITE_FILE_PATH -c $CONFIG_FILE -p nfs
    ```

3. Run the following command to import values after they are created by using either the user interface or API.

    ```
    SQLITE_FILE_PATH=/tmp/example.sqlite
    CONFIG_FILE=$(pwd)/harvester/config.ini
    python3 harvester tags $SQLITE_FILE_PATH -c $CONFIG_FILE
    ```

4. Use `-p nfs` option if you want to import NFS-based metadata.

    ```
    SQLITE_FILE_PATH=/tmp/example.sqlite
    CONFIG_FILE=$(pwd)/harvester/config.ini
    python3 harvester metaocean $SQLITE_FILE_PATH -c $CONFIG_FILE -p nfs
    ```

Configuring the Harvester

The `config.ini` file inside the harvester directory provides critical support in the data import behavior. The following are the available variables to configure:

BlockSizeBytes
: It indicates the block size configured in the datasource that is being imported. Normally, the value is set to 4096 bytes, which helps to calculate the real capacity allocated on the filesystem based on the blocks required to store the file.

AvailableMOVW
: It must be set to 10 as it is the only supported value for optimal data ingestion. It helps the harvester to calculate the pieces of a set of records that can be divided for parallel insertion.

MaxProcessingWorkers

It is an ignored variable, and use a one worker process always. If a higher level of parallelism is required, then run another harvester to process an additional SQLite file.

IngestRequestRetries
: After the harvester finishes processing the metadata from the SQLite file, it sends it using an HTTP request to the import service for it to be indexed in the database. But it is possible to fail to do so for different reasons, such as a temporary network unavailability or a considerable API overload blocking incoming requests. By default, the harvester tries to send the metadata 120 times, depending on the highest value and the time limit. The suggested range is 120 to 240.

StatusRequestRetries
: After the ingest request is sent, the harvester follows up with additional requests to get the status of the ingest operation to confirm whether it finished successfully. It might take a long time, so retrying is required. The suggested range is 240 to 360.

TagsWorkdir
: The name of the directory that is presented inside the harvester directory is where temporary files are stored during tag harvesting. By default, `tmp` is the name of this directory, but it can be updated in case it is necessary to have a more descriptive directory name.

MOBatchSize and TagsBatchSize
: Number of rows to read from the SQLite and load into memory at a time, a higher value is associated to a higher memory usage. It is not recommended to use a value higher than 1,00,000 because it does not represent a performance benefit. The suggested values are 10000, 50000 and 100000.

MOBatchesPerSegment and TagsBatchesPerSegment
: Number of batches to process before sending them using an ingestion request, a higher value is associated with bigger files. Using a value higher than 100 does not translate to a direct performance benefit. The suggested values are 10, 50 and 100.

Data ingestion system requirements

Ensure that you must have the following system requirements:

- 4 CPU cores and 16GB of RAM.
- At least 1.5x the size of the SQLite file to import must be available as free space on the filesystem. For example, if the SQLite file is 1GB, then make sure at least 1.5GB of storage is available.
- Python 3.11 or higher.

Installation
: Run the following commands to download and install IBMFDC.
Optionally, you can set up a virtual environment to benefit from an isolated installation.

  Note: Before proceeding, ensure you have an active and authenticated session in the OpenShift console. This is necessary for the `oc` command to retrieve route information correctly. For more information, see Red Hat Documentation.

```
DCS_NS="ibm-data-cataloging"
DCS_IMPORT_SVC_HOST=$(oc -n ${DCS_NS} get route isd-import-service -o jsonpath='{.spec.host}')

curl -O -J "https://${DCS_IMPORT_SVC_HOST}/v1/ibmfdc" --insecure
WHEEL_FILE=$(ls ibmfdc-*.whl)
pip3 install ${WHEEL_FILE}
rm -f ${WHEEL_FILE}
```

Preparing the CLI configuration file
: By default, the ibmfdc command looks for a file named config.ini in the current working directory. If you want to specify a different configuration file, use the `oc` command-line argument.
Configuration file format:

```
ConsoleRouteURL = <DCS_CONSOLE_URL>
ImportCSVRouteURL = <DCS_IMPORT_SVC_URL>
AdminUsername = <DCS_USERNAME>
AdminPassword = <DCS_PASSWORD>
```

  For example:

```
ConsoleRouteURL = https://console-ibm-data-cataloging.apps.<domain>
ImportCSVRouteURL = https://isd-import-service-ibm-data-cataloging.apps.<domain>
AdminUsername = sdadmin
AdminPassword = Passw0rd
```

Ingesting remote metadata using the IBMFDC Harvester
: The IBMFDC Harvester supports ingesting both base and additional metadata simultaneously. However, all necessary tags must be created beforehand, otherwise the meta table in the SQLite file will be ignored. This is because none of its columns matches any existing tag names.
Importing base and additional metadata:

```
ibmfdc harvester -m meta <path_to_sqlite_file>
```

  Importing only base metadata (ignores tags):

```
ibmfdc harvester <path_to_sqlite_file>
```

FKEY migration script

The FKEY migration script is a fix to avoid potential rare collisions in the FKEY file identifier when ingesting billions of records.

About this task

The FKEY migration script fix consists of a script that needs to be run in the Db2 terminal of IBM Data Cataloging.

Procedure

1. Run the following command to do SSH to the `c-Db2u-0 pod`.

   ```
   oc -n ibm-data-cataloging rsh c-isd-db2u-0
         su - db2inst1
   ```

2. Create a file named FKEY_Updater.sh with the FKEY migration script.
   The script iterates over a list of data sources, fixes the FKEY in all the files, and ingests them in each data source of the previous IBM Data Cataloging service versions.

   ```
         # ========================================================
         # How to invoke
         # Send the list of datasources to fix as follows:
         # ./FKEY_Updater.sh datasource1 datasource2 datasource3
         # ========================================================

         start_date=`date +%s.%N`
         datasourcearray=( "$@" )

         echo "Connecting to database..."
         db2 connect to bludb
         echo "Connected to BLUDB."
         printf '\n'
         echo "List of datasources to update:"
         printf ' - %s\n' "${datasourcearray[@]}"
         printf '\n'

         for datasource in "${datasourcearray[@]}"
         do
           echo "Starting update procedures for datasource '${datasource}'..."
           printf '\n'
           echo "Updating table ACESMAPLOADBASE..."
           db2 "update bluadmin.acesmaploadbase amlb set amlb.fkey=(mo.cluster || '_' || mo.datasource || '_' || mo.inode)
   from bluadmin.metaocean mo where amlb.fkey=mo.fkey and mo.datasource='${datasource}';"
           echo "ACESMAPLOADBASE table updated successfully."
           printf '\n'
           echo "Updating table ACOGMAPLOADBASE..."
           db2 "update bluadmin.acogmaploadbase acmlb set acmlb.fkey=(mo.cluster || '_' || mo.datasource || '_' || mo.inode)
   from bluadmin.metaocean mo where acmlb.fkey=mo.fkey and mo.datasource='${datasource}';"
           echo "ACOGMAPLOADBASE table updated successfully."
           printf '\n'
           echo "Updating table ACESMAP (This action could take several minutes)..."
           db2 "update bluadmin.acesmap am set am.fkey=(mo.cluster || '_' || mo.datasource || '_' || mo.inode) from
   bluadmin.metaocean mo where am.fkey=mo.fkey and mo.datasource='${datasource}';"
           echo "ACESMAP table updated successfully."
           printf '\n'
           echo "Updating table ACOGMAP (This action could take several minutes)..."
           db2 "update bluadmin.acogmap acm set acm.fkey=(mo.cluster || '_' || mo.datasource || '_' || mo.inode) from
   bluadmin.metaocean mo where acm.fkey=mo.fkey and mo.datasource='${datasource}';"
           echo "ACOGMAP table updated successfully."
           printf '\n'
           echo "Updating table METAOCEAN (This action could take several minutes)..."
           db2 "update bluadmin.metaocean mo set mo.fkey=(mo.cluster || '_' || mo.datasource || '_' || mo.inode) where not
   REGEXP_LIKE(mo.fkey, mo.cluster || '_' || mo.datasource || '_' || mo.inode) and mo.datasource='${datasource}';"
           echo "METAOCEAN table updated successfully."
           printf '\n'
           echo "Updates on datasource '${datasource}' done successfully."
           printf '\n'
         done
         printf '\n'
         end_date=`date +%s.%N`
         runtime=$(echo "$end_date - $start_date" | bc -l)

         echo "Execution time was $runtime seconds."
   ```

3. Run the script.

   ```
   ./FKEY_Updater.sh datasource1 datasource2 datasource3
   ```

   The final execution time of the script depends on the number of files that are ingested in the database.

Configurable Db2 log trimmer

Db2 log trimmer tool provides a mechanism to trim the informative logs that are generated by scanning a data source. It uses the Harvester CLI to ingest data into the IBM Data Cataloging.

Configurable Db2 log trimmer tool helps to better handle the filesystem usage by deleting informative files placed in the Db2 PVC.

From IBM Fusion 2.8.x release, the capabilities of the Db2 log trimmer are extended by a `ConfigMap` that gives an option of setting the currency for running the trimmer (in minutes). It selects and deletes the old files and `.BAD` extension files.
Important:

- The Db2 `.BAD` files are log files that are generated by failed transactions in Db2 when trying to ingest data.
- If there is failure at the Db2 level that requires the guidance of a Db2 expert, those `.BAD` files can be indispensable to perform the root cause analysis and it is recommended NOT TO DELETE those files unless needed.

The fields in the `ConfigMap` are the following:

- **time-limit-min** - default value is "720" (files with more than 720 minutes of last modification)
- **bad-file-erase** - default value is "0" (not to delete the `.BAD` files)
- **frequency-min** - default value is "720" (run the log trimmer each 720 minutes)

You can edit the values of the `ConfigMap` by using the following command.

```
oc edit configmap db2-log-trimmer -n ibm-data-cataloging
```

IBM Data Cataloging service installation and upgrade issues

Use this troubleshooting information to resolve install and upgrade problems that are related to the IBM Data Cataloging service.

Installation stuck because of resource quota

Problem statement

The IBM Data Cataloging installation is stuck because of resource quota set on IBM Fusion and IBM Data Cataloging namespaces. tuntil both the resource quota were removed

Resolution

Set the resource quota values for IBM Data Cataloging as follows:

```
kind: ResourceQuota
apiVersion: v1
metadata:
  name: dcs-resource-quota
  namespace: ibm-data-cataloging
spec:
  hard:
    limits.cpu: '100'
    limits.memory: 180Gi
    pods: '120'
    requests.cpu: '20'
    requests.memory: 40Gi
```

Data cataloging upgrade stuck for more than 1 hour

Problem statement

IBM Data Cataloging upgrade is stuck for more than 1 hour and the progress shows 15%.

Symptoms

The IBM Data Cataloging upgrade runs for more than 1 hour and gets stuck at 15%. On the OpenShift® Console, the installed operator shows just the IBM Data Cataloging operator and not **`IBM DB2`** and **`Streams for Apache Kafka`** operators under the **`ibm-data-cataloging`** namespace.
Also, the Subscription for the IBM Data Cataloging operator shows `InstallPlanPending, Reason "RequiresApproval`.

Resolution

1. In the OpenShift console, go to Operators > Installed Operators tab.
2. Select IBM Storage Discover operator.
3. Go to Subscriptions tab, and select InstallPlan.
4. Approve the InstallPlan.
   There is an alternative method using OpenShift CLI as follows:
   a. Run the following command to get the InstallPlan that is not approved.

      ```
      oc get ip -n ibm-data-cataloging -o=jsonpath='{.items[?(@.spec.approved==false)].metadata.name}'
      ```

   b. Run the following command to get the InstallPlan name.

      ```
      oc patch installplan $(oc get ip -n ibm-data-cataloging -o=jsonpath='{.items[?
      (@.spec.approved==false)].metadata.name}') -n ibm-data-cataloging --type merge --patch '{"spec":
      {"approved":true}}'
      ```

5. Wait for the upgrade process to complete.

Data Cataloging import service pod in CrashLoopBackOff state

Diagnosis

1. Check for CrashLoopBackOff on the import service pod.

   ```
   oc -n ibm-data-cataloging get pod -l role=import-service
   ```

2. Confirm that the logs show `permission denied` error:

   ```
   oc -n ibm-data-cataloging logs -l role=import-service
   ```

Resolution

1. Debug import service pod.

   ```
   oc -n ibm-data-cataloging debug deployment/isd-import-service --as-user=0
   ```

2. Update the directory permissions.

```
chmod 775 /uploads
mkdir -p /uploads/failed_requests
chmod 775 /uploads/failed_requests
exit
```

IBM Data Cataloging database schema job is not in a completed state during installation or upgrade

Note: This procedure is applicable to recover DB2 that goes into an unavailable mode after the service installation and related to a degraded state of the IBM Data Cataloging service.

Symptoms

The `isd-db2whrest` or `isd-db-schema` pods report a not ready or error state.
Run the following command to view the common logs:

```
oc -n ibm-data-cataloging logs -l 'role in (db2whrest, db-schema)' --tail=200
```

Go through the logs to check whether the following error exists:

```
Waiting on c-isd-db2u-engn-svc port 50001...

db2whconn - ERROR - [FAILED]: [IBM][CLI Driver] SQL1224N The database manager is not able to accept new requests, has
terminated all requests in progress, or has terminated the specified request because of an error or a forced interrupt.
SQLSTATE=55032

Connection refused
```

Resolution

1. Restart Db2:

```
oc -n ibm-data-cataloging rsh c-isd-db2u-0
sudo wvcli system disable -m "Disable HA before Db2 maintenance"
su - ${DB2INSTANCE}
db2stop
db2start
db2 activate db BLUDB
exit
sudo wvcli system enable -m "Enable HA after Db2 maintenance"
```

2. Confirm that Db2 HA-monitoring is active:

```
sudo wvcli system status
exit
```

3. Check whether the problem occurred during installation or upgrade or a post installation problem.
4. If this occurs during an upgrade or installation, recreate the `isd-db-schema` job and monitor the pod until it gets to a completed state.

```
SCHEMA_OLD="isd-db-schema-old.json"
SCHEMA_NEW="isd-db-schema-new.json"
oc -n ibm-data-cataloging get job isd-db-schema -o json > $SCHEMA_OLD
jq 'del(.spec.template.metadata.labels."controller-uid") | del(.spec.selector) | del (.status)' $SCHEMA_OLD >
$SCHEMA_NEW
oc -n ibm-data-cataloging delete job isd-db-schema
oc -n ibm-data-cataloging apply -f $SCHEMA_NEW

oc -n ibm-data-cataloging get pod | grep isd-db-schema
```

5. If this is a post installation problem, restart db2whrest.

```
oc -n ibm-data-cataloging delete pod -l role=db2whrest
```

IBM Data Cataloging installation is stuck for more than 1 hour

Note: This procedure must be used only during installation problems, not for upgrades or any subsequent issues.

Symptoms

The IBM Data Cataloging installation running for more than an hour, and it stuck somewhere between 35% and 80% (both inclusive).

Resolution

1. Run the following command to scale down the operator.

```
oc -n ibm-data-cataloging scale --replicas=0 deployment/spectrum-discover-operator
```

2. Run the following command to scale down workloads.

```
oc -n ibm-data-cataloging scale --replicas=0 deployment,statefulset -l component=discover
```

3. Run the following command to remove the DB schema job if present.

```
oc -n ibm-data-cataloging delete job isd-db-schema --ignore-not-found
```

4. Run the following commands to delete the Db2 instance and the password secret.

```
oc -n ibm-data-cataloging delete db2u isd
oc -n ibm-data-cataloging delete secret c-isd-ldapblueadminpassword --ignore-not-found
```

5. Wait until the Db2 pods and persistent volume claims get removed.

```
oc -n ibm-data-cataloging get pod,pvc -o name | grep c-isd
```

6. Run the following command to scale up the operator.

```
oc -n ibm-data-cataloging scale --replicas=1 deployment/spectrum-discover-operator
```

IBM Data Cataloging service is not installed successfully on IBM Fusion HCI with GPU nodes

Problem statement

The data catalog service is in installing state for hours.

Resolution

To resolve the problem, do the following steps:

1. Patch FSD with new affinity to not schedule `isd` workload on those nodes:

```
oc -n <Fusion_namespace> patch fusionservicedefinitions.service.isf.ibm.com data-cataloging-service-definition  --
patch "$(cat fsd_dcs_patch.yaml)"
```

The fsd_dcs_patch.yaml file:

```
cat >> fsd_dcs_patch.yaml << EOF

apiVersion: service.isf.ibm.com/v1
kind: FusionServiceDefinition
metadata:
  name: data-cataloging-service-definition
  namespace: <Fusion_namespace>
spec:
  onboarding:
    parameters:
      - dataType: string
        defaultValue: ibm-data-cataloging
        descriptionCode: BMYSRV00003
        displayNameCode: BMYSRV00004
        name: namespace
        required: true
        userInterface: false
      - dataType: storageClass
        defaultValue: ''
        descriptionCode: BMYDC0300
        displayNameCode: BMYDC0301
        name: rwx_storage_class
        required: true
        userInterface: true
      - dataType: bool
        defaultValue: 'true'
        descriptionCode: descriptionCode
        displayNameCode: displayNameCode
        name: doInstall
        required: true
        userInterface: false
      - dataType: json
        defaultValue: '{"accept": true}'
        descriptionCode: descriptionCode
        displayNameCode: displayNameCode
        name: license
        required: true
        userInterface: false
      - dataType: json
        defaultValue: '{"nodeAffinity":{"requiredDuringSchedulingIgnoredDuringExecution":{"nodeSelectorTerms":
[{"matchExpressions":[{"key":"nvidia.com/gpu","operator":"NotIn","values":["Exists"]}]}]}}}'
        descriptionCode: descriptionCode
        displayNameCode: displayNameCode
        name: affinity
        required: true
        userInterface: false

EOF
```

If the output shows this error message `Error from server (UnsupportedediaType): the body of the request was in an unknown format - accepted media types include: application/json-patch+json, application/merge-patch+j son, application/apply-patch+yaml`, then you need to follow the steps to resolve this issue:

a. Go to OpenShift Container Platform web console.
b. Go to Operators > Installed Operators tab, under `<Fusion_namespace>`, select the IBM Fusion operator.
c. Select the IBM Fusion service instance tab, and select the `data-cataloging-service-instance`.
d. Select the YAML tab, and edit the YAML file for the `data-cataloging-service-instance`. Under `spec.onboarding.parameters`, ensure you add the following lines.

```
- dataType: json
          defaultValue: '{"nodeAffinity":{"requiredDuringSchedulingIgnoredDuringExecution":
{"nodeSelectorTerms":[{"matchExpressions":[{"key":"isf.ibm.com/nodeType","operator":"NotIn","values":
["gpu"]}]}]}}}'
          descriptionCode: descriptionCode
          displayNameCode: displayNameCode
          name: affinity
          required: true
          userInterface: false
```

2. Display the patch FSD:

```
oc -n <Fusion_namespace> get fusionservicedefinitions.service.isf.ibm.com data-cataloging-service-definition -o yaml
```

3. Install from the user interface.

IBM Data Cataloging database pods stuck during initialization phase

Symptoms

Multiple Db2 pods use the host port 5002, which can cause pods to stay in the `init` phase.

Resolution

1. Uninstall the IBM Data Cataloging service. For procedure, see Uninstalling IBM Data Cataloging.
2. Create a file with the IBM Data Cataloging `FusionServiceDefinition` patch.

```
cat >> fsd_dcs_patch.yaml << EOF
apiVersion: service.isf.ibm.com/v1
kind: FusionServiceDefinition
metadata:
  name: data-cataloging-service-definition
  namespace: ibm-spectrum-fusion-ns
spec:
  onboarding:
    parameters:
      - dataType: string
        defaultValue: ibm-data-cataloging
        descriptionCode: BMYSRV00003
        displayNameCode: BMYSRV00004
        name: namespace
        required: true
        userInterface: false
      - dataType: storageClass
        defaultValue: ''
        descriptionCode: BMYDC0300
        displayNameCode: BMYDC0301
        name: rwx_storage_class
        required: true
        userInterface: true
      - dataType: bool
        defaultValue: 'true'
        descriptionCode: descriptionCode
        displayNameCode: displayNameCode
        name: doInstall
        required: true
        userInterface: false
      - dataType: json
        defaultValue: '{"accept": true}'
        descriptionCode: descriptionCode
        displayNameCode: displayNameCode
        name: license
        required: true
        userInterface: false
      - dataType: json
        defaultValue: '{"size":1,"mln":2,"storage":{"activelogs":{"requests":"300Gi"},"data":
{"requests":"600Gi"},"meta":{"requests":"100Gi"},"activelogs":{"tempts":"100Gi"}}}'
        descriptionCode: descriptionCode
        displayNameCode: displayNameCode
        name: dbwh
        required: true
        userInterface: false
EOF
```

3. Apply the patch to downsize the Db2 cluster.

```
oc -n ibm-spectrum-fusion-ns patch fusionservicedefinitions.service.isf.ibm.com data-cataloging-service-definition --
type=merge --patch-file fsd_dcs_patch.yaml
```

4. Install IBM Data Cataloging service from the IBM Fusion user interface. For procedure, see Installing IBM Data Cataloging.

IBM Fusion IBM Data Cataloging known issues

List of all issues in the IBM Data Cataloging service along with its resolution.

The following known issues exist in IBM Data Cataloging service, with workarounds included wherever possible. If you come across an issue that cannot be solved by using these instructions, contact IBM support .

Db2 SQL1218N: No pages currently available in the buffer pool

Problem statement

Db2 database encounters SQL1218N error indicating insufficient memory in the buffer pool.

Resolution

Update the `Db2uCluster` memory limit in OpenShift® by following these steps:

1. Log in to the OpenShift cluster.

```
oc login --token=<OC_TOKEN> --server=<OC_CLUSTER>
```

Purpose: Authenticates to the OpenShift cluster by using a token.

- `<OC_TOKEN>`: Your OpenShift authentication token (obtain from OpenShift web console).
- `<OC_CLUSTER>`: The OpenShift cluster API server URL (for example: https://api.cluster.example.com:6443).

2. Get the `Db2uCluster` resource name.

```
db2ucluster_name=$(oc get Db2uCluster -n ibm-data-cataloging -o name)
```

Purpose: Retrieves the `Db2uCluster` resource name and stores it in a variable.

- `-n ibm-data-cataloging`: Specifies the namespace where `Db2uCluster` is deployed.
- `-o name`: Outputs only the resource name (format: `db2ucluster/name`).
- Stores the result in `db2ucluster_name` variable for use in subsequent commands.

3. Patch the memory limit to 20Gi.

```
oc patch Db2uCluster ${db2ucluster_name} -n ibm-data-cataloging --type=merge -p '{"spec":{"podConfig":{"db2u":
{"resource":{"db2u":{"limits":{"memory":"20Gi"}}}}}}}'
```

Purpose: Updates the memory limit for Db2u pods to 20Gi.

- `${db2ucluster_name}`: Uses the variable from step 2.
- `--type=merge`: Merges the patch with existing configuration.
- `-p`: JSON patch that updates `spec.podConfig.db2u.resource.db2u.limits.memory` to 20Gi.

4. Wait for `Db2uCluster` to be ready.

```
oc wait --for=jsonpath='{.status.state}'=Ready ${db2ucluster_name} -n ibm-data-cataloging --timeout=600s
```

Purpose: Waits until the `Db2uCluster` reaches Ready state after the memory update.

- `--for=jsonpath='{.status.state}'=Ready`: Waits for the `status.state` field to be Ready.
- `--timeout=600s`: Maximum wait time of 10 minutes (600 seconds)
- The command blocks until ready or timeout is reached.

Resource quota limitation

With resource quota set on IBM Fusion and IBM Data Cataloging namespaces, the IBM Data Cataloging installation gets stuck until both the resource quota are removed.

Cannot access the IBM Data Cataloging user interface

Problem statement
: The IBM Data Cataloging user interface becomes inaccessible.

Cause
: If the IBM Data Cataloging user interface is not available, then check whether the IBM Data Cataloging service is in healthy state or not.

Resolution
: For steps to resolve this issue, see Fusion Data Foundation is in a degraded state.

Live events stopped working after IBM Data Cataloging

Problem statement
: Live events functionality stopped updating record events after the IBM Data Cataloging is upgraded.

Resolution
: Follow the steps to resolve this issue:

- Delete the watcher on IBM Storage Scale without deleting the connection on the IBM Data Cataloging service as follows:
  1. Run the following command to list the watchers.

```
/usr/lpp/mmfs/bin/mmwatch all list
```

Example output:

```
Filesystem testing has 1 watcher(s):
Cluster ID            WatchID/PID     Type       Start Time               Path                       Description
-------------------------------------------------------------------------------------------------------------------
13748412159870826098   CLW1733515374   FSYS       Fri Dec 06 12-02-55 2024  /ibm/testing
```

  2. Run the following command to delete the watchers.

```
sudo /usr/lpp/mmfs/bin/mmwatch testing disable --watch-id CLW1733515374
```

Example output:

```
[I] Successfully checked or deleted Clustered Watch policy skip partitions for watch: CLW1733515374
[I] Successfully removed Clustered Watch policy rules for watch: CLW1733515374
[I] Successfully removed Clustered Watch configuration file from the CCR for watch: CLW1733515374
[I] Successfully removed Clustered Watch configuration directory for watch: CLW1733515374
[I] Successfully checked or removed Clustered Watch global catchall and config fileset skip partitions for watch: CL
[I] Successfully disabled Clustered Watch: CLW1733515374
```

  3. Run the command again to list the watchers to verify the watcher deletion.

```
/usr/lpp/mmfs/bin/mmwatch all list
```

Example output:

```
Cluster ID       WatchID/PID   Type    Start Time        Path           Description
 ------------------------------------------------------------------------------------------------------------------------
Filesystem testing has no watchers.
```

4. Run the following command to export cluster's URL and IBM Storage Scale working directory as environmental variables.

```
export CLUSTER_URL = "yourcluster.ibm.com"
export WD_PATH="/example/path/wd"
```

5. Run the following command to manually add the watcher on the IBM Storage Scale.

```
sudo /usr/lpp/mmfs/bin/mmwatch testing enable --event-handler kafkasink --sink-brokers "isd-kafka-tlsext-
bootstrap-ibm-data-cataloging.apps.qa-bm-$CLUSTER_URL:443" --sink-topic scale-le-connector-topic --sink-auth-
config $WD_PATH/kafka/auth.file --events
IN_ATTRIB,IN_CLOSE_WRITE,IN_MODIFY,IN_CREATE,IN_DELETE,IN_MOVED_FROM,IN_MOVED_TO
```

Example output:

```
[I] Beginning enablement of Clustered Watch with newly created watch ID: CLW1733516766
[I] Verified the watch type is FSYS for filesystem testing
[I] Successfully added Clustered Watch configuration file into CCR for watch: CLW1733516766
[I] Successfully added Clustered Watch configuration in the Spectrum Scale file system: testing for watch: CLW173351
[I] Successfully added Clustered Watch policy rules for watch: CLW1733516766
[I] Successfully checked or created Clustered Watch global catchall and config fileset skip partitions for watch: CL
[I] Successfully checked or created Clustered Watch policy skip partitions for watch: CLW1733516766
[I] Successfully enabled Clustered Watch: CLW1733516766
```

6. Run the command again to list the watchers to verify the watcher creation.

```
/usr/lpp/mmfs/bin/mmwatch all list
```

Example output:

```
Filesystem testing has 1 watcher(s):
 Cluster ID       WatchID/PID   Type    Start Time        Path           Description
 ------------------------------------------------------------------------------------------------------------------------
 13748412159870826098  CLW1733516766  FSYS    Fri Dec 06 12-26-07 2024 /ibm/testing
```

7. On IBM Data Cataloging, re-scan the IBM Storage Scale connection.
8. Any modifications on the scanned IBM Storage Scale folder should be monitored through live events and automatically updated on IBM Data Cataloging.

IBM Data Cataloging service in Metro-DR setup shows in Degraded state

Diagnosis

1. IBM Data Cataloging service is in degraded state
2. Run the following command to check whether the Pod **isd-db2whrest** is not ready:

```
oc -n ibm-data-cataloging get pod -l role=db2whrest
```

3. Run the following command to check whether Db2 retries the network check and fails because of the timeout:

```
oc -n ibm-data-cataloging logs -l type=engine --tail=100
```

Example output:

```
+ timeout 1 tracepath -l 29 c-isd-db2u-1.c-isd-db2u-internal
+ [[ 17 -lt 120 ]]
+ (( n++ ))
+ echo 'Command failed. Attempt 18/120:'
Command failed. Attempt 18/120:
```

Resolution

Increase the time before timeout, usually change from 1 second to 3-5 seconds.

1. Modify the timeout from 1 to 3 in **isd-db2u-0**:

```
oc -n ibm-data-cataloging exec c-isd-db2u-0 -- sudo sed -i 's/timeout 1 tracepath/timeout 3 tracepath/g'
/db2u/scripts/include/common_functions.sh
```

2. Wait until the current attempt exceeds the predefined 120 retries. After it restarts, it picks the updated value:

```
oc -n ibm-data-cataloging logs -l type=engine --tail=50
```

3. Monitor **db2whrest** pod readiness:

```
oc -n ibm-data-cataloging get pod -l role=db2whrest -w
```

COS connection reporting scan aborted due to inactivity

If a COS connection scan fails with the error “Scan aborted because of a long period of inactivity”, it can be resolved by editing the settings file connections/cos/scan/scanner-settings.json within the data PV and choosing a higher value for notifier_timeout than the default value of 120 seconds. The change will be picked on the next scan. No pod restart is required.

Database connection issue after reboot

If an unexpected cluster update or node reboot causes database connection issues. For resolution, see the steps mentioned in the *IBM Data Cataloging database schema job is not in a completed state during installation or upgrade* section. .

Image pull error due to authentication failure

Problem statement

The OpenShift Container Platform login token expires occasionally, and as this is the container image registry password, this breaks the service account access to the registry.

Resolution

If a pod is failing to pull an image from the registry with an authentication error, then re-create the image-registry-pull-secret and relink the service accounts to the new secret:

```
oc delete secret image-registry-pull-secret
HOST=$(oc get route default-route -n openshift-image-registry --template='{{ .spec.host }}')
oc create secret docker-registry image-registry-pull-secret \
    --docker-server="${HOST}" \
    --docker-username=kubeadmin \
    --docker-password="$(oc whoami -t)"
for account in spectrum-discover-operator strimzi-cluster-operator spectrum-discover-ssl-
zookepper spectrum-discover-sasl-zookeeper; do oc secrets link $account image-registry-pull-
secret --for=pull; done
```

Visual query builder search terms overrides SQL search when going into individual mode

If a search is started in the query builder, then changed to SQL mode, the initial group search is as expected but if expanded to individual records it uses the query builder terms as the base. A workaround is to clear the visual query before changing to SQL query.

LDAPS configuration failing if dollar sign is in password

Currently, the dollar sign is not supported on passwords for ldaps configuration. A workaround is to create a password without the dollar sign in it.

Content search policy missing files

If there are issues with the incorrect expected data count while running a policy, you must verify that the connection is active, and rescan to get the latest data ingested to IBM Data Cataloging. After successful upgrade of IBM Data Cataloging, a rescan of existing connections is recommended.

REST API returns token with unprintable characters

It is a noted issue that a carriage return (\r) is included at the end of HTTP response headers due to an issue with curl. This has been known to occasionally break scripts that use an auth token from the IBM Data Cataloging appliance as shown here:

```
$ curl -k -H "Authorization: Bearer ${TOKEN}" https://$SDHOST/policyengine/v1/tags
curl: (92) HTTP/2 stream 0 was not closed cleanly: PROTOCOL_ERROR (err 1)
```

As such, it is recommended to filter out the \r character. If you have a line like the following in bash:

```
`TOKEN=$(curl -i -k https://$SDHOST/auth/v1/token -u "$SDUSER:$SDPSWD" | grep -i x-auth-token |
awk '{print $2}')`
```

Add a | tr -d '\r' at the end to avoid running into this issue:

```
`TOKEN=$(curl -i -k https://$SDHOST/auth/v1/token -u "$SDUSER:$SDPSWD" | grep -i x-auth-token |
awk '{print $2}' | tr -d '\r')`
```

Adding S3 connection gives false negative

Problem statement

When a connection of type S3 is added through IBM Data Cataloging user interface, it gives an undefined error message.

Resolution

Refreshing the browser removes the error message, and the connections table shows that the S3 connection was successful.

Scale datamover AFM and ILM capabilities not working properly due to SDK misleading function when deploying an application

Problem statement

When deploying IBM Data Cataloging service deployments pods `scaleafmdatamover` and `scaleilmdatamover` might show an error on logs when application is deployed.

For example:

```
2023-07-20 02:51:54,311 - ibm_spectrum_discover_application_sdk.ApplicationLib - INFO - Invoking conn manager at
http://172.30.255.202:80/connmgr/v1/internal/connections
Traceback (most recent call last):
File "/application/ScaleAFMDataMover.py", line 1023, in
APPLICATION = ScaleAFMApplicationBase(REGISTRATION_INFO)
File "/application/ScaleAFMDataMover.py", line 112, in init
self.conn_details = self.get_connection_details()
File "/usr/local/lib/python3.9/site-packages/ibm_spectrum_discover_application_sdk/ApplicationLib.py", line 492, in
get_connection_details
raise Exception(err)
```

```
UnboundLocalError: local variable 'err' referenced before assignment
2023-07-20 02:51:54,367 INFO exited: scaleafm-datamover (exit status 1; not expected)
2023-07-20 02:51:55,368 INFO gave up: scaleafm-datamover entered FATAL state, too many start retries too quickly
```

Cause

SDK bug when deploying applications on DCS, causing deployed applications not to behave properly and pods in the incorrect state to show errors.

Resolution

After identified this behavior, then follow the steps to resolve this issue:

1. Verify the connmgr API is running and accessible through HTTP (curl to the connmgr service would be enough).
2. Check the application pod and remove it to be redeployed.

Policies are not finished, resulting in a hanging state

Problem statement

The policies are not finished, which results in a hanging state.

Cause

Inconsistent behavior on policies results in a no finish status. The issue is still under investigation.

Resolution

Identify the policy engine pod and eliminate it; OpenShift Container Platform will create another pod automatically, and policies will be executed and finished properly after the pod creation.
Example:

```
oc -n ibm-data-cataloging delete pod -l role=policyengine
```

IBM Data Cataloging service goes degraded state after IBM Fusion HCI rack restart

Problem statement

The IBM Data Cataloging service goes degraded after some of the nodes are restarted or after IBM Fusion HCI rack restart. Several pods go pending with the following errors `Unable to attach or mount volumes: unmounted volumes=[spectrum-discover-db2wh], unattached volumes=[], failed to process volumes=[]: timed out waiting for the condition` and `MountVolume.SetUp failed for volume "xxx" : rpc error: code = Internal desc = staging path yyy for volume zzz is not a mountpoint`.

Resolution

Run the following steps to resolve this issue:

1. Run the following commands to make the compute nodes unschedule and drain them one after one.

   ```
   oc adm cordon worker4.fusion-test-zlinux.cp.fyre.ibm.com
   oc adm drain worker4.fusion-test-zlinux.cp.fyre.ibm.com --ignore-daemonsets --force --delete-emptydir-data
   ```

2. After the node is drained, ensure it schedules again and then proceed to another node with the same process.
   It removes stale directory entries from nodes that are detected as mount points.

3. The issue automatically resolves and IBM Data Cataloging service must be in a healthy state after all the nodes are backed up.

Db2 license does not display correctly on the upgrade set up

Problem statement

The Db2 license displays incorrectly on the IBM Data Cataloging service upgrade set up.

Resolution

Run the following steps to resolve this issue:

1. Run the following command to get the subsequent IBM Data Cataloging scoped content.

   ```
   oc project ibm-data-cataloging
   ```

2. For the new license to take effect, delete the Db2 engine pods for the **Db2uCluster** or **Db2uInstance**:

   ```
   oc delete $(oc get po -l type=engine,formation_id=isd -oname)
   ```

3. After the new Db2 pod is ready, verify the updated Db2 license:

   ```
   oc exec -it  c-isd-db2u-0 -- su - db2inst1 -c "db2licm -l"
   ```

   For more about Db2 community edition license certificate key, see Upgrading your Db2 Community Edition license certificate key

ConstraintsNotSatisfiable for db2u-operator

Problem statement

The Db2 version used by IBM Data Cataloging service 2.1.6 got removed from the latest version of the IBM Operator Catalog.

Resolution

Follow the steps to resolve the issue:

1. Add or update the IBM Catalog Source object **ibm-operator-catalog** in the **openshift-marketplace** namespace during install or upgrade.

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: ibm-operator-catalog-old
   ```

```
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:@sha256:c2538264cb1882b1c98fea5ef162f198ce38ed8c940e82e3b9db458a9a46cb15
  updateStrategy:
    registryPoll:
      interval: 45m
```

2. Update the `ibm-operator-catalog` source with the latest tag after the installation or upgrade completed.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog-old
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

ConstraintsNotSatisfiable for db2u-operator

Problem statement

The Db2 version used by IBM Data Cataloging service 2.1.6 got removed from the latest version of the IBM Operator Catalog.

Resolution

Follow the steps to resolve the issue:

1. Add or update the IBM Catalog Source object `ibm-operator-catalog` in the `openshift-marketplace` namespace during install or upgrade.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog-old
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog@sha256:5d606e4eb2b875e0b975f892e80343105ea5fb0d67f96e1400d77a715f6df72a
 updateStrategy:
    registryPoll:
      interval: 45m
```

2. Update the `ibm-operator-catalog` source with the latest tag after the installation or upgrade completed.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog-old
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

Content-Aware Storage (CAS)

The IBM Fusion Content-Aware Storage service is engineered to enhance the value of customers' AI applications, such as retrieval augmented generation (RAG) for faster time to insights, reduced cost, improved performance, enhanced security, and streamlined operations.

CAS provides turn-key processing of unstructured data for use in multi-modal RAG applications using IBM Docling Multimodal services and NVIDIA NeMo Retriever Extraction microservices. It supports automated extraction of text, tables, and charts from Enterprise documents and makes the extracted information available for use in RAG applications. NVIDIA GPUs are used to accelerate the data extraction processing. For more information on CAS-related frequently asked questions, see Content-Aware Storage FAQs.

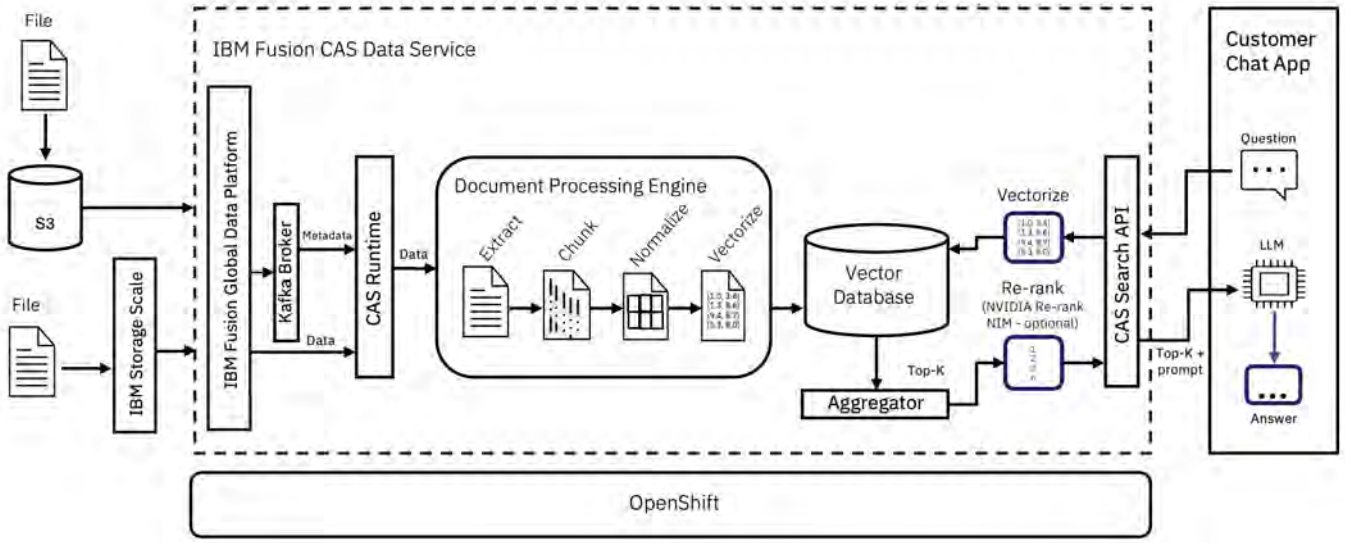
CAS uses IBM Storage Scale to cache existing Enterprise data residing on S3 sources to optimize processing with the IBM Docling Multimodal services and NVIDIA NeMo Retriever Extraction microservices without having to manage multiple copies of data. CAS also supports automated processing of data residing in the IBM Storage Scale filesystem. Incremental change processing is used to detect when data is modified in the IBM Storage Scale filesystem and automatically process the changed data for use in RAG applications.

- **Architecture**
  The objective of the Content-Aware Storage (CAS) architecture is to facilitate smooth interactions between LLMs and extensive quantities of unstructured data, amplifying insight derivation and suggestion functionalities.
- **Installing Content-Aware Storage (CAS)**
  Initiate the installation of CAS from the user interface of IBM Fusion. It installs Kafka, CAS operator, and CNPG.

- **Configuring Content-Aware Storage (CAS)**
  This section provides instructions for configuring CAS.
- **Client application with Content-Aware Storage (CAS)**
  You are required to provide a client application that uses the CAS search API to use the data ingested by CAS as part of your RAG application. This topic outlines the workflow for querying the CAS database using the exposed API.
- **Monitoring**
  This page provides a comprehensive view of monitoring capabilities, structured into two key sections: monitoring using dashboard in Content-Aware Storage (CAS) user interface and metric collection through Prometheus.
- **Uninstalling Content-Aware Storage (CAS) service**
  When you do not need CAS service, uninstall it from IBM Fusion.
- **Troubleshooting issues and known limitations in Content-Aware Storage (CAS)**
  The common troubleshooting steps and limitations in CAS service.
- **Related information**
  IBM publications, IBM MediaCenter videos, and external videos that contain information that is related to Content-Aware Storage (CAS).

Architecture

The objective of the Content-Aware Storage (CAS) architecture is to facilitate smooth interactions between LLMs and extensive quantities of unstructured data, amplifying insight derivation and suggestion functionalities.



The IBM Fusion CAS service uses IBM Fusion Global Data Platform service to remote mount a IBM Storage Scale cluster. This cluster can be Storage Scale software-based, or Storage Scale Server 3500 or 6000-based. CAS automatically processes files that are stored on the remote IBM Storage Scale cluster into parsed content, and metadata is normalized and stored along with vector embeddings in its managed vector database. Document processing is available with IBM's Docling Multimodal services or NVIDIA NeMo Retriever Extraction microservices.

IBM Storage Scale watch folders are used to detect changed data in the remote IBM Storage Scale filesystem, which triggers ingest processing by CAS of just the changed content.

Files stored in external S3 data sources are ingested through Active File Management (AFM) running in Global Data Platform. No external IBM Storage Scale filesystem is required. CAS performs a scan of the AFM-S3 fileset automatically to determine the data that needs to be ingested.

The CAS service provides a search API that aggregates semantic and keyword search to return the most accurate results. NVIDIA's reranking is an optional search component that can be added to provide more accuracy.

Customer applications such as a chatbot can send natural language searches to the CAS search API. Upon receiving the request, CAS creates a vector representation of the natural language search and performs a query against the vector database that uses the search method that is specified in the API (semantic, keyword, or hybrid). In the NVIDIA Multimodal pipeline, the top 100 results are sent to a CAS aggregator component that uses additional keyword and optionally vendor reranking to produce the top number of requested results. The optimized results are passed to a reranking function, which performs an additional optimization of the results before returning them to the search API.

In a customer provided RAG client, the CAS search API can be used to retrieve relevant search results, which can then be combined with the original prompt or question and sent to a large language model (LLM) to generate a response for the end user. IBM provides a sample RAG client in the form of a chatbot that customers can use to explore CAS capabilities or as a foundation for developing their own chatbot or RAG-based solution.

Note: CAS uses natural language processing (NLP) and AI technologies to extract semantic meaning from only unstructured text data.
Important: To know about the resource usage like, vCPU, memory, storage, and GPU, see System requirements.

- **Understanding user personas**
  The personas of IBM Fusion Content-Aware Storage (CAS) service include Scale Admin, Data Engineer, and Developer. The Developer connects RAG application for working with the ingested content.

Understanding user personas

The personas of IBM Fusion Content-Aware Storage (CAS) service include Scale Admin, Data Engineer, and Developer. The Developer connects RAG application for working with the ingested content.

Scale Admins

Scale Admins are individuals responsible for managing storage software, such as IBM Storage Scale clusters and storage infrastructure. The Scale Admin deploys CAS with IBM Fusion and AI pipeline onto OpenShift® Container Platform cluster. The Scale Admin can see events for CAS in the same interface as IBM Fusion for a consistent serviceability experience.

The key responsibilities of a Scale Admin are as follows:

- Deploy, configure, and manage IBM Storage Scale clusters to ensure optimal performance and reliability.
- Configure and fine-tune IBM Storage Scale to efficiently handle high-throughput, low-latency AI data processing, ensuring integration with GPUs and AI frameworks.
- Work with Data Engineer to oversee data ingestion, preprocessing, and movement across storage tiers.
- Monitor system health of the Scale environment, implement caching strategies, and optimize data placement to prevent bottlenecks in AI model training and deployment.
- Use automation tools and scripts to streamline data lifecycle management, scaling storage resources dynamically to meet AI workload demands.

Data Engineer

Data engineers use data lakehouses as a flexible and scalable solution to store, process, and analyze diverse data types. Data Engineer defines a fileset on its storage content and connects it to an AI pipeline. In addition, the Data Engineer monitors the execution of the AI pipeline to understand the amount and timeliness of processing changes in content. As a Data Engineer, I can configure an Nvidia NIM pipeline to my Scale Fileset through CRs.

The key responsibilities of a Data Engineer are as follows:

- Storage of raw data in its original format. Including structured, semi-structured and unstructured data.
- Design and implement data ingestion pipelines that extract data from various sources, ensuring that all relevant data is available via a single interface for analysis and reporting.
- Maintain/curate metadata catalogs that capture information about data sources, data transformations, and data dependencies.
- Transform, augment, cleanse, and derive new datasets from existing data in accordance with their consumer's needs.
- Define access controls, data security measures, and data privacy policies to ensure compliance with policy and regulatory requirements.
- Enable data scientists, analysts, and other stakeholders to explore and analyze data.

Application Developer

The Application Developer queries and chats using the client enterprise application.

Installing Content-Aware Storage (CAS)

Initiate the installation of CAS from the user interface of IBM Fusion. It installs Kafka, CAS operator, and CNPG.

Before you begin

- Proceed with the installation only after all the prerequisites are met. For more information about prerequisites, see Planning and prerequisites.
- Ensure that you have storage class and is marked as default. For example, **`ibm-spectrum-fusion`**. For more information about how to mark a storage class as default, see Prerequisites topic.
- Ensure all Scale projects and pods are running.
- For IBM Fusion services and platform support matrix, see Support matrix for IBM Fusion versions.

About this task

Use this procedure to install the CAS service from the Services page of the IBM Fusion user interface. You can also install the CAS service using CLI. For more information about CLI usage, see Configuring Content-Aware Storage (CAS) through CLI.

Follow the procedure if:

- Your data resides on an external IBM Storage Scale filesystem.
- You prefer to use an external IBM Storage Scale filesystem as cache filesystem for your S3 data source.

Otherwise, follow the procedure in Installing CAS without a remotely mounted IBM Storage Scale filesystem.

Procedure

1. In the IBM Fusion user interface, go to Services page.
2. In the Available section, click the CAS tile.
3. In the CAS page, go through the features and capabilities of the service and click Install.
   If the pre-check raises a warning, then resolve it at OpenShift level.
   The installation starts and a notification appears on the Services page. You can see the progress of the installation in the Services > **Installed** section. After the installation is completed successfully, you can see the status as Healthy. A success notification gets displayed. It disappears automatically after some time.

   From the ellipsis menu, you can download logs and view documentation.
4. In the Install service window, select storage class and click Install.

5. Validate the installation.
    - Verify the CAS installation from the IBM Fusion user interface:
      After you enable the CAS service, you can view the service version and health status. From the ellipsis menu, you can download logs and view documentation. After you successfully collect the logs, a success notification is displayed. The notification disappears automatically after some time.
    - Verify the CAS installation from the OpenShift® Container Platform web console:
        - In the OpenShift console, go to Home > Projects.
        - In the Projects page, search and check for the availability of a namespace with string `ibm-cas`.
        - Go to Operators > Installed Operators and check for the availability of CAS.
        - Open Content Aware Storage and check whether a CAS service definition is available.
        - After you enable the CAS service, you can view the service version and health status. From the ellipsis menu, you can download logs and view documentation. After you successfully collect the logs, a success notification is displayed. The notification disappears automatically after some time.

What to do next

- To verify the percentage completion of prerequisites post installation, do the following steps:
    1. Open IBM Fusion user interface.
    2. Go to Content-Aware Storage > Overview.
    3. In the Jump right in section, click Getting started with CAS tile.
    4. A side panel opens with the status of the following prerequisites:
        - Integrate NVIDIA NIM
        - Set up of Kafka Broker
        - Connect caching file system
- After you successfully install CAS, set up a Scale user to enable watch creation, configure domains and data sources. Map the domain to the data source. For more information about these procedures, see Configuring Content-Aware Storage (CAS).

- **Planning and prerequisites**
  Go through the Content-Aware Storage (CAS) installation prerequisites before you install the service.
- **Installing CAS without a remotely mounted IBM Storage Scale filesystem**
  Install CAS on bare metal without a remotely mounted IBM Storage Scale filesystem using local Data Foundation storage and S3 buckets as the data source.
- **Installing NVIDIA NeMo Retriever Library**
  The Content-Aware Storage (CAS) Document Processor Engine can either use IBM Docling Multimodal or NVIDIA NeMo Retriever Extraction for document extraction.
- **Configuring NVIDIA Inference Microservices (NIMs)**
  After you install CAS, the **`cas-config`** configmap must be created. It provides the NeMo Retriever Library endpoint information.
- **Configuring a Multi-Instance GPU (MIG) with CAS**
  Multi-Instance GPU (MIG) partitions a single NVIDIA GPU into multiple isolated instances. Each instance operates as an independent GPU, enabling resource sharing across multiple deployments.
- **Configuring IBM Storage Scale user to enable watch creation**
  Configure an IBM Storage Scale user with all the needed privileges to enable watch creation.

Planning and prerequisites

Go through the Content-Aware Storage (CAS) installation prerequisites before you install the service.

General requirements

Ensure that you meet the following prerequisites for the CAS solution:

- Ensure that you have IBM Fusion 2.13.0 version.
  Important:
    - CAS service installation can be performed only if IBM Fusion is installed by using the default **`ibm-spectrum-fusion-ns`** namespace. If the IBM Fusion installation is performed by using a custom namespace, the CAS installation cannot be completed.
    - IBM Fusion Access for Storage Area Network (SAN), Global Data Platform, and CAS are mutually exclusive services. If IBM Fusion Access for SAN is planned or installed, you cannot deploy the Global Data Platform or CAS services. Review your architectural requirements and deployment goals before installing these services.
- Resource requirements
- * IBM Storage Scale requirements. See IBM Storage Scale requirements.
- * Configure Scale remote filesystems. See Configure Scale remote file systems.
- * Ensure that you have the latest IBM Storage Scale 5.2.3.x version.
- * After at least one IBM Storage Scale remote filesystem is configured, in the OpenShift® console, go to Storage > Storage Class and verify whether the storage class created from the IBM Storage Scale remote filesystem is available for CAS. For example, if you enter **`ibm-spectrum-fusion`**, verify whether it is available.
- Configure NIMs. See Configuring NVIDIA Inference Microservices (NIMs).

* Skip these requirements if you are installing CAS without a remotely mounted IBM Storage Scale filesystem.

Configuring flags in CAS ConfigMap

To use the NVIDIA re-ranker service, you can add the **`NVMM_NEMO_RANKER`** and **`NVMM_NEMO_RANKER_SERVICE`** flags to the cas-config ConfigMap.

For more information, see Configuring NVIDIA Inference Microservices (NIMs).

Allowlist for proxy

Add the following URLs to the allowlist for proxy installation:

- **apigee.googleapis.com**: Using Apigee Hybrid runtime, you can learn about proxies, shared flows, and other key components. It also provides information about configuration and system health.
- **apigeeconnect.googleapis.com**: Needed for **apigee-mart-server** and **apigee-connect** communication when **VPC-SC** is enabled.
- **binaryauthorization.googleapis.com**: Optional, only for Anthos if the binary authorization is enabled.
- **gcr.io**: Google Container Registry where the container images are hosted.
- **raw.githubusercontent.com**: To install the operator manifest.

Resource requirements

Work with IBM representative to ensure you have all the hardware components that are necessary for the CAS solution. Allocate three OpenShift compute nodes.

- The following resources are required for CAS to run on the three compute nodes:
  - 3 x NVIDIA L40S or H100 GPUs
  - An additional L40S or H100 GPU is needed for the optional re-ranker service.
  - 166 vCPU (83 physical cores)
  - 1024GB memory
  - 1TB IBM Storage Scale
  - Most common IBM Storage Scale ESS configuration is 48 NVMe drives with the capacity of 30TB in the ESS 6000 with 4 CX-7 network adapters (1.5 TB memory).
  - When using the Docling Multimodal Processing Engine, see System requirements.
- For CAS service system requirements, see System requirements for each of the components in addition to the resource requirements.
  Note: Content-Aware Storage (CAS) depends upon IBM Fusion management software and the Global Data Platform service.

Enabling GPUs in OpenShift

To enable GPUs in the OpenShift cluster, following components must be installed to support GPU workloads:

- Red Hat's Node Feature Discovery Operator - See the Red Hat documentation.
- NVIDIA's GPU operator version - See the NVIDIA documentation.

To deploy GPU operators in a disconnected or airgapped environment, see the NVIDIA documentation.

NVIDIA Multimodal Document Processing Engine requirements

- The NVIDIA GPU operator, multi-modal RAG blueprint, and NVIDIA NIM operator are pulled from external NVIDIA NGC registry. As part of this process, you must provide NVIDIA NGC license keys to pull the NVIDIA components.
  NVIDIA NIM is a set of optimized cloud-native microservices that are designed to simplify the deployment of generative AI models anywhere, across cloud, data center, and GPU-accelerated workstations. Initially, you must deploy the NVIDIA NeMo Retriever. For more information about the procedure to deploy, see NVIDIA documentation.

- Optional: NVIDIA text reranking:
  The NVIDIA text reranker is an optional component that requires an additional supported GPU. This component takes the initial set of search results and uses additional scoring from NVIDIA to return the top ranked results. For more information about deploying NeMo Retriever text reranking NIM, see NVIDIA documentation.

- Contact IBM for the following details:
  - For optimization and performance of L40S or H100 GPU utilization of the NVIDIA NeMo Retriever Library multimodal PDF Ingest blueprint.
  - To configure NeMo Retriever Library NIM components with Multi-Instance GPU (MIG) for optimal use of supported GPUs or with time slicing. For more information, see Configuring a Multi-Instance GPU (MIG) with CAS.
  - Additional tuning parameters required to optimize pipeline performance.

Compatibility matrix for NVIDIA's NeMo Retriever Library Multimodal blueprint with CAS

Content-Aware Storage (version)	NVIDIA NeMo Retriever Library Multimodal blueprint
v1.0.6, v1.0.7, v1.1.0	v25.9.0
v1.1.1, v1.1.2, v1.1.3, v1.1.4	v26.1.2
v1.1.5 and newer	v26.3.0

IBM Storage Scale requirements

If your datasource is on S3:

- Active File Management (AFM) is required to be configured on IBM Storage Scale remote file system. For more information, see Configuring Active File Management.
- IBM Storage Scale cluster must be at version 5.2.3.1 or above.

Configure Scale remote file systems

To configure remote file systems, see Connecting to remote IBM Storage Scale file systems. For software requirement levels, see Software requirement levels.
Important:

- When you setup the Global Data Platform user to create the connection to the remote file system, the user must be part of either the **CsiAdmin** or **ContainerOperator** groups.
  However, these groups do not have enough privileges to allow watcher creation. As a prerequisite to enable successful watcher creation, add these users to the **StorageAdmin** group. Create a user with all the privileges needed as a prerequisite. For more information about this configuration, see Configuring IBM Storage Scale user to enable watch creation.

- After the remote file system installation is complete, mark your storage class, for example, **ibm-spectrum-fusion**, as default, by running the following command:

```
oc patch storageclass ibm-spectrum-fusion -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

Offline installation

Content-Aware Storage supports usage of the Docling Multimodal document processing engine in an offline environment. For more information about the procedure to deploy offline, see Mirroring Content-Aware Storage (CAS) images.

Installing CAS without a remotely mounted IBM Storage Scale filesystem

Install CAS on bare metal without a remotely mounted IBM Storage Scale filesystem using local Data Foundation storage and S3 buckets as the data source.

Before you begin

- Proceed with the installation only after all the prerequisites are met. For more information about prerequisites, see Planning and prerequisites.
- For IBM Fusion services and platform support matrix, see Support matrix for IBM Fusion versions.

Supported environment

- IBM Fusion Bare Metal
- Data Foundation is installed or intended to be used as the local storage provider.
- Data resides on AWS or IBM S3 buckets only.
- OpenShift® Hosted Control Plane (HCP) clusters

About this task

To install CAS without a remotely mounted IBM Storage Scale filesystem, follow the instructions in README.md.

Limitations

- The automation script creates a local IBM Storage Scale filesystem named **`cache-fs`**. This filesystem is reserved exclusively as a cache for S3 data and must not be used by other applications.
- The **`cache-fs`** filesystem is not expandable.
- If your CAS data source is connected but I/O ingestion is slow, check whether:
  1. Data Foundation shows a warning alert **`SystemMemoryExceedsReservation`**.
  2. Any Object Storage Daemon (OSD) pod has **`restarts`**:

     ```
     oc get pods -n openshift-storage -l app=rook-ceph-osd
     ```

     For pods that have **`restarts`**, check if termination reason is **`OOMKilled`**:

     ```
     oc describe pod rook-ceph-osd-xxx -n openshift-storage
     ```

     Use the following procedure to increase the OSD pod memory from the default 8 GB:

     https://access.redhat.com/solutions/6959127
- The system supports the shutdown or failure of only one quorum node at any specific time.

Installing NVIDIA NeMo Retriever Library

The Content-Aware Storage (CAS) Document Processor Engine can either use IBM Docling Multimodal or NVIDIA NeMo Retriever Extraction for document extraction.

Note: NVIDIA Ingest (nv-ingest) has been renamed to NeMo Retriever Library.
For more information about installing the GPU operator, see Enabling GPUs in OpenShift.

For more information about installing NVIDIA NeMo Retriever Library, see the NVIDIA documentation.

Verify the PodPidsLimit

To ensure compatibility and stability, you must update your cluster and deployment settings when using NeMo Retriever Library v26.3.0 with CAS v1.1.5 or newer.

1. To retrieve the current configuration, run the following command:

   ```
   oc get KubeletConfig
   ```

2. To increase the pod PID limit, modify the **`kubeletConfig`** by adding the following **`podPidsLimit`** to the **`spec`** section:

   ```
   spec:
     kubeletConfig:
       podPidsLimit: 12228
   ```

   Note: After updating the configuration, all worker nodes in the cluster get restarted which can cause a temporary disruption.
   For more information on changing the **`podPidsLimit`** when using the ROSA environment, see the Red Hat documentation.

Install the NIM Operator

NeMo Retriever Library requires that the NIM Operator is deployed and running. It is required to install the NIM Operator first before running the NeMo Retriever Library Helm chart. The operator can be installed from the OpenShift® marketplace. Search for The NVIDIA NIM Operator for Kubernetes, and install using the default options.

Select NVIDIA Inference Microservices (NIMs)

Before installing NIMs from the Helm chart, select the NIMs that are needed to extract your documents. For example, if your documents include audio files, installation of the audio NIM is required. Each NIM requires one GPU resource, which can either be a full GPU, or a GPU slice using a Multi-Instance GPU (MIG). The audio, reranking, and image captioning NIMs require a full GPU to run properly. By default, the NeMo Retriever Library Helm chart deploys five NIMs as indicated in the following table. For more information about MIG, see Configuring a Multi-Instance GPU (MIG) with CAS.

NIM	Usage	Parameter in Helm chart	Default setting
`llama-nemotron-embed-1b-v2`	Text embedding	`nimOperator.embedqa.enabled`	`true`
`nemotron-page-elements-v3`	Object detection within documents	`nimOperator.page_elements.enabled`	`true`
`nemotron-graphic-elements-v1`	Chart extraction	`nimOperator.graphic_elements.enabled`	`true`
`nemotron-table-structure-v1`	Table extraction	`nimOperator.table_structure.enabled`	`true`
`nemotron-ocr-v1`	Optical character recognition (OCR)	`nimOperator.ocr.enabled`	`true`
`nemotron-nano-12b-v2-v1`	Vision language model (VLM) image captioning	`.nemotron_nano_12b_v2_v1.enabled`	`false`
`audio`	Audio extraction	`nimOperator.audio.enabled`	`false`
`llama-nemotron-rerank-1b-v2`	Reranker for query accuracy	`nimOperator.rerankqa.enabled`	`false`

Install NeMo Retriever Library

To install NeMo Retriever Library, follow these steps:

1. Create a new project called `nv-ingest`:

   ```
   oc new-project nv-ingest
   ```

2. In your command line session, set the following bash variables:

   ```
   NGC_API_KEY=<api key>
   NAMESPACE=nv-ingest
   ```

   Tip: To generate an NVIDIA NGC API key, see Generate Your NGC Keys.

3. Add `helm` repositories:

   ```
   # Nvidia nemo-microservices NGC repository
   helm repo add nemo-microservices https://helm.ngc.nvidia.com/nvidia/nemo-microservices --username='$oauthtoken' --
   password=<NGC_API_KEY>

   # Nvidia NIM NGC repository
   helm repo add nvidia-nim https://helm.ngc.nvidia.com/nim/nvidia --username='$oauthtoken' --password=<NGC_API_KEY>

   # NVIDIA NIM Base repository
   helm repo add nim https://helm.ngc.nvidia.com/nim --username='$oauthtoken' --password=<NGC_API_KEY>

   # Nvidia NIM baidu NGC repository
   helm repo add baidu-nim https://helm.ngc.nvidia.com/nim/baidu --username='$oauthtoken' --password=<NGC_API_KEY>
   ```

4. Deploy NeMo Retriever Library using the `helm` chart. To select NIMs, see Select NVIDIA Inference Microservices (NIMs).
   Example 1: Deploy NeMo Retriever Library with the five default NIM services deployed

   ```
   helm upgrade \
      --install \
      nv-ingest \
      https://helm.ngc.nvidia.com/nvidia/nemo-microservices/charts/nv-ingest-26.3.0.tgz \
      -n ${NAMESPACE} \
      --username '$oauthtoken' \
      --password "${NGC_API_KEY}" \
      --set ngcImagePullSecret.create=true \
      --set ngcImagePullSecret.password="${NGC_API_KEY}" \
      --set ngcApiSecret.create=true \
      --set ngcApiSecret.password="${NGC_API_KEY}" \
      --set image.repository="nvcr.io/nvidia/nemo-microservices/nv-ingest" \
      --set image.tag="26.3.0" \
      --set milvusDeployed=false
   ```

   Example 2: Deploy NeMo Retriever Library with the default NIM services (same as Example 1) along with the audio Riva NIM

   ```
   helm upgrade \
      --install \
      nv-ingest \
      https://helm.ngc.nvidia.com/nvidia/nemo-microservices/charts/nv-ingest-26.3.0.tgz \
      -n ${NAMESPACE} \
      --username '$oauthtoken' \
      --password "${NGC_API_KEY}" \
      --set ngcImagePullSecret.create=true \
      --set ngcImagePullSecret.password="${NGC_API_KEY}" \
      --set ngcApiSecret.create=true \
      --set ngcApiSecret.password="${NGC_API_KEY}" \
      --set nimOperator.audio.enabled=true \
      --set image.repository="nvcr.io/nvidia/nemo-microservices/nv-ingest" \
      --set image.tag="26.3.0" \
      --set milvusDeployed=false
   ```

Example 3: Deploy NeMo Retriever Library with the default NIM services (same as example 1) along with the audio Riva NIM and the reranking NIM

```
helm upgrade \
   --install \
   nv-ingest \
   https://helm.ngc.nvidia.com/nvidia/nemo-microservices/charts/nv-ingest-26.3.0.tgz \
   -n ${NAMESPACE} \
   --username '$oauthtoken' \
   --password "${NGC_API_KEY}" \
   --set ngcImagePullSecret.create=true \
   --set ngcImagePullSecret.password="${NGC_API_KEY}" \
   --set ngcApiSecret.create=true \
   --set ngcApiSecret.password="${NGC_API_KEY}" \
   --set nimOperator.audio.enabled=true \
   --set nimOperator.rerankqa.enabled=true \
   --set image.repository="nvcr.io/nvidia/nemo-microservices/nv-ingest" \
   --set image.tag="26.3.0" \
   --set milvusDeployed=false
```

Example 4: Minimum configuration

For a minimum configuration installation that extracts only text from files (text or PDF), the text embedding NIM is required. However, the other NIM services can be disabled.

```
helm upgrade \
   --install \
   nv-ingest \
   https://helm.ngc.nvidia.com/nvidia/nemo-microservices/charts/nv-ingest-26.3.0.tgz \
   -n ${NAMESPACE} \
   --username '$oauthtoken' \
   --password "${NGC_API_KEY}" \
   --set ngcImagePullSecret.create=true \
   --set ngcImagePullSecret.password="${NGC_API_KEY}" \
   --set ngcApiSecret.create=true \
   --set ngcApiSecret.password="${NGC_API_KEY}" \
   --set nimOperator.graphic_elements.enabled=false \
   --set nimOperator.page_elements.enabled=false \
   --set nimOperator.table_structure.enabled=false \
   --set nimOperator.ocr.enabled=false \
   --set image.repository="nvcr.io/nvidia/nemo-microservices/nv-ingest" \
   --set image.tag="26.3.0" \
   --set milvusDeployed=false
```

5. Check the PVC sizes created for the `NIMCache`:

   ```
   oc get pvc -n nv-ingest
   ```

   Note: You are recommended to increase the audio NIM to 250Gi, reranker NIM to 100Gi, and the `embedqa` NIM to 100Gi to avoid issues downloading the required models.
   Expand the following PVCs:
   a. `embedqa` NIM PVC

      ```
      oc patch pvc llama-nemotron-embed-1b-v2-pvc  -n nv-ingest --type merge -p '{"spec":{"resources":{"requests":
      {"storage":"100Gi"}}}}'
      ```

   b. Audio NIM PVC (only applicable when the audio NIM is enabled)

      ```
      oc patch pvc audio-pvc  -n nv-ingest --type merge -p '{"spec":{"resources":{"requests":{"storage":"250Gi"}}}}'
      ```

   c. Reranker NIM PVC (only applicable when the reranking NIM is enabled):

      ```
      oc patch pvc llama-nemotron-rerank-1b-v2-pvc  -n nv-ingest --type merge -p '{"spec":{"resources":{"requests":
      {"storage":"100Gi"}}}}'
      ```

6. Verify that the pods are running:

   ```
   oc get deployments -n nv-ingest
   ```

   Example output:

   ```
   NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
   audio                               1/1     1            1           5d23h
   llama-nemotron-embed-1b-v2          1/1     1            1           5d23h
   llama-nemotron-rerank-1b-v2         1/1     1            1           5d23h
   nemotron-graphic-elements-v1        1/1     1            1           5d23h
   nemotron-nano-12b-v2-vl             1/1     1            1           5d23h
   nemotron-ocr-v1                     1/1     1            1           5d23h
   nemotron-page-elements-v3           1/1     1            1           5d23h
   nemotron-table-structure-v1         1/1     1            1           5d23h
   nv-ingest                           1/1     1            1           5d18h
   nv-ingest-opentelemetry-collector   1/1     1            1           5d18h
   nv-ingest-zipkin                    1/1     1            1           5d18h
   ```

Configuring NVIDIA Inference Microservices (NIMs)

After you install CAS, the `cas-config` configmap must be created. It provides the NeMo Retriever Library endpoint information.

To determine the NeMo Retriever Library endpoint information, you must know the namespace in which NeMo Retriever Library is installed. The endpoints are constructed by using the pattern `http(s)://..svc.cluster.local`. When you use the default installation namespace of NeMo Retriever Library and because it does not support `https` by default, it results in the following URLs:

- Embed service URL: `http://llama-32-nv-embedqa-1b-v2.nv-ingest.svc.cluster.local`
- Ingest service URL: `http://nv-ingest.nv-ingest.svc.cluster.local`

Note: These URLs must not end with the `/` character.
To complete this activity, use the **oc** command line tool and run the following command:

```
oc apply -f - <<EOF
apiVersion: v1
kind: ConfigMap
metadata:
  name: cas-config
  namespace: ibm-cas
data:
  NVMM_EMBED_SERVICE: <Embed Service URL as determined above>
  NVMM_NIM_SERVICE:  <Ingest Service URL as determined above>
EOF
```

NVIDIA re-ranker

Optionally, you can set **`NVMM_NEMO_RANKER`** in the **`CasInstall`** CR to enable the NVIDIA re-ranker service. This service analyzes semantic relevance and reorders search results to improve precision in enterprise search and streamline AI-driven workflows. For more information, see the NVIDIA documentation.

- Re-ranker service URL: `http://llama-32-nv-rerankqa-1b-v2.nv-ingest.svc.cluster.local`

Note: The service URL must not end with the `/` character.
To enable the NVIDIA re-ranker service, set the **`NVMM_NEMO_RANKER`** flag to **`YES`** and the **`NVMM_NEMO_RANKER_SERVICE`** flag to the re-ranker service URL in the **`cas-config`** ConfigMap:

```
spec:
oc patch configmap cas-config \
  -n ibm-cas \
  --type merge \
  -p '{"data":{"NVMM_NEMO_RANKER": "YES", "NVMM_NEMO_RANKER_SERVICE": "<Re-ranker service URL as determined above>"}}'
```

Configuring a Multi-Instance GPU (MIG) with CAS

Multi-Instance GPU (MIG) partitions a single NVIDIA GPU into multiple isolated instances. Each instance operates as an independent GPU, enabling resource sharing across multiple deployments.

Before you begin

MIG support requirements

Only specific NVIDIA GPUs support MIG. For more information on the supported models and their configurations, see the NVIDIA documentation.
Note: The NeMo Retriever Library NIMs support only a subset of the GPUs that support MIG.

Before you configure MIG, plan how to partition each GPU by using the supported profiles. Each profile defines the amount of memory and compute engines that the slice can use.

The following table shows the minimum recommended MIG profile for each NIM service by GPU model:

NIM service	H100	H200	RTX PRO 6000
`nemotron-embed`	`mig-7g.80gb`	`mig-7g.141gb`	`mig-1g.24gb`
`nemotron-page-elements`	`mig-1g.10gb`	`mig-1g.18gb`	`mig-1g.24gb`
`nemotron-graphic-elements`	`mig-1g.10gb`	`mig-1g.18gb`	`mig-1g.24gb`
`nemotron-table-structure`	`mig-1g.10gb`	`mig-1g.18gb`	`mig-1g.24gb`
`nemotron-ocr`	`mig-1g.20gb`	`mig-1g.18gb`	`mig-1g.24gb`
`audio`	`mig-7g.80gb`	`mig-7g.141gb`	`mig-4g.96gb`
`nemotron-nano`	`mig-7g.80gb`	`mig-7g.141gb`	

For more information about each of these NIM services, see Select NVIDIA Inference Microservices (NIMs).

For further examples, see NVIDIA MIG User Guide.

Example: An RTX PRO 6000 GPU can be partitioned in the following way:

- One 96 GB slice with 4 compute engines
- Two 48 GB slices with 2 compute engines each

Each RTX PRO 6000 GPU contains four compute engines. Ensure that the total number of compute engines across all slices on a GPU do not exceed four.
Note: For optimal ingestion performance, assign a full GPU MIG profile to the `nemotron-embed` NIM service to increase GPU compute capacity.

About this task

This task configures MIG on NVIDIA GPUs that support the MIG feature. The steps involve enabling MIG, defining GPU partitions, applying the MIG configuration to your nodes, and assigning deployment to specific GPU slices.

To configure MIG on NVIDIA GPUs, perform the following steps:

Procedure

1. Apply the following configmap in the `nvidia-gpu-operator` namespace:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nv-ingest-mig
data:
  config.yaml: |
    version: v1
    mig-configs:
      all-disabled:
        - devices: all
          mig-enabled: false
      nv-ingest-mig-config:
        - devices: [0, 1]
          mig-enabled: true
          mig-devices:
            "2g.20gb": 2
            "3g.40gb": 1
```

2. Temporary stop any pods that use the GPUs. The MIG manager cannot reconfigure GPUs that are in use.
3. Patch the `gpu-cluster-policy` and set the MIG strategy value to "mixed".

```
kubectl patch clusterpolicies.nvidia.com/gpu-cluster-policy \
    --type='json' \
    -p='[{"op":"replace", "path":"/spec/mig/strategy", "value":"mixed"}]'
```

4. Label one or more nodes with the GPUs that you want to configure your MIG settings, referencing the MIG configuration defined in the configmap. Then set the correct value to the `NODE` variable:

```
NODE=ip-10-0-9-79.us-west-2.compute.internal
MIG_CONFIGURATION=nv-ingest-mig-config
oc label node $NODE nvidia.com/mig.config=$MIG_CONFIGURATION --overwrite
```

5. Monitor the pods in the `nvidia-gpu-operator` namespace. The configuration must recycle several pods, including the `gpu-feature-discovery` pods.
6. To verify the MIG configuration, run the following command:

```
nvidia-smi
```

You can see MIG slices with memory usage metrics after NeMo Retriever Library is activated. Initially, the memory usage appears as 1 MiB until the pods start to use the slices.

```
+-----------------------------------------------------------------------------------------+
| MIG devices:                                                                            |
+------------------+--------------------------------+-----------+-----------------------+
| GPU  GI  CI  MIG |                     Memory-Usage |        Vol|         Shared        |
|      ID  ID  Dev |                       BAR1-Usage | SM     Unc| CE ENC  DEC  OFA  JPG |
|                  |                                |        ECC|                       |
|==================+================================+===========+=======================|
|  0    2   0   0  |            3590MiB / 40192MiB  | 42      0 |  3   0    2    0    0 |
|                  |                  2MiB / 32767MiB  |           |                       |
+------------------+--------------------------------+-----------+-----------------------+
|  0    3   0   1  |            2959MiB / 19968MiB  | 28      0 |  2   0    1    0    0 |
|                  |                  2MiB / 16383MiB  |           |                       |
+------------------+--------------------------------+-----------+-----------------------+
|  0    4   0   2  |            2971MiB / 19968MiB  | 28      0 |  2   0    1    0    0 |
|                  |                  2MiB / 16383MiB  |           |                       |
+------------------+--------------------------------+-----------+-----------------------+
|  1    2   0   0  |           25386MiB / 40192MiB  | 42      0 |  3   0    2    0    0 |
|                  |                  2MiB / 32767MiB  |           |                       |
+------------------+--------------------------------+-----------+-----------------------+
|  1    3   0   1  |            2955MiB / 19968MiB  | 28      0 |  2   0    1    0    0 |
|                  |                  2MiB / 16383MiB  |           |                       |
+------------------+--------------------------------+-----------+-----------------------+
|  1    4   0   2  |            2281MiB / 19968MiB  | 28      0 |  2   0    1    0    0 |
|                  |                  2MiB / 16383MiB  |           |                       |
+------------------+--------------------------------+-----------+-----------------------+
```

Note: If the MIG configuration fails to apply, retrigger the MIG manager by disabling and re-enabling it in the cluster policy configuration:

```
migManager:
    config:
      default: all-disabled
      name: nv-ingest-mig
    enabled: true -> false -> true
```

7. Before you scale up your microservices that require GPUs, such as NeMo Retriever Library NIMs, you must add the following snippets to their deployments:
   - For larger NIMs, such as the NVIDIA re-ranker:

     ```
     containers:
             - resources:
                 limits:
                   nvidia.com/mig-3g.40gb: '1'
     ```

   - For smaller NIMs:

     ```
     containers:
             - resources:
                 limits:
                   nvidia.com/mig-2g.20gb: '1'
     ```

8. Scale up the deployments. The pods are then allocated to the correct GPU slice.

Configuring IBM Storage Scale user to enable watch creation

Configure an IBM Storage Scale user with all the needed privileges to enable watch creation.

IBM Storage Scale has a spectrum of user roles, each with varying degree of privileges. For example, Security Administrator, Administrator, System Administrator, Storage Administrator, CSI Administrator, Container Operator. For more information about the user roles, see Scale Administration Guide.

As part of doing the IBM Storage Scale remote mount, the Global Data Platform asks the user to ensure that they are a part of both CSI Administrator and Container Operator. The StorageAdmin is needed to enable CAS to manage filesystem configurations such as clustered watch. For more information about the configuration steps, see Connecting to remote IBM Storage Scale file systems.

It is not sufficient for the CAS user to be a part of the Storage Administrator group, in addition to the **`CsiAdmin`**, **`ContainerOperator`** groups. As a IBM Storage Scale admin, run the following example command from command line:

```
/usr/lpp/mmfs/gui/cli/chuser csi-storage-gui-user -g CsiAdmin,ContainerOperator,StorageAdmin
```

Example output:

```
EFSSG0020I The user cast-user has been successfully changed.
EFSSG1000I The command completed successfully.
```

It involves the alignment of the CSI user name to match the one previously created during the remote cluster setup by using the Global Data Platform.

Configuring IBM Storage Scale for change detection

Determine your IBM Storage Scale version first:

```
mmdiag --version
```

For IBM Storage Scale 6.0.0.x and later, no manual configuration is required.

If you are running IBM Storage Scale version 5.2.3.x, you need to manually enable S3 change detection. As an IBM Storage Scale administrator, run the following commands:

```
mmchconfig syncReadWFConfig=yes --force -i
mmchconfig afmPtrashOpt=3 -i
```

Note: Run these commands only once for the entire IBM Storage Scale cluster.

Configuring Content-Aware Storage (CAS)

This section provides instructions for configuring CAS.

Do the following step to configure CAS from the IBM Fusion user interface:

- Creating a data source or Creating a domain and connecting with data source

Tip: Starting from CAS v1.1.1, CAS installation automatically configures the Kafka broker connection on IBM Storage Scale.
You can also configure CAS service using CLI. For more information about CLI usage, see Configuring Content-Aware Storage (CAS) through CLI.
Note: Before you begin the configuration, ensure you have met all the prerequisites and installed CAS successfully. For the procedure to install, see Installing Content-Aware Storage (CAS).
Quick start to configuration:
Note: If the Getting started with CAS section is not available, it indicates that all the prerequisites are not met.

1. In the IBM Fusion user interface, go to Content-Aware Storage > Overview.
2. In the Jump right in section, you can click the following tiles to quick start configuration to streamline data preparation, chunking and delivery to embedding or inference systems:
    - Connect a data source
    - Create a domain
    - View Search APIs
3. To view the percentage completion of the configuration, do the following steps:
    a. In the Jump right in section, click Getting started with CAS tile.
    b. A side panel opens with the status of the following Data ingestion:
        - Connect a data source
        - Create a domain
        - View Search APIs

- **Integrating CAS and IBM Data Cataloging**
  Rule-based integration between IBM Data Cataloging and Content-Aware Storage (CAS) ensures selective, policy-driven data ingestion into CAS vector stores.
- **Creating a data source**
  A data source is a repository or a system that stores and provides data for use in applications or systems. They are storage locations that provide data to use in AI workflows.
- **Creating a domain and connecting with data source**
  Domains collect, transform, and transfer data from multiple data sources to a destination, enabling efficient data flow for analysis and AI models.
- **Configuring Document Processor Engine**
  Steps to configure Document Processor Engines: IBM Docling Multimodal and NVIDIA Multimodal.
- **Configuring Content-Aware Storage (CAS) through CLI**
  You can use the CRs to configure CAS.

- **Using your own Hugging Face access token with CAS for NVIDIA Multimodal**
  NVIDIA NeMo Retriever Library uses models that are downloaded from Hugging Face to do token-based splitting on the content extracted from documents. NVIDIA recommends to use the meta-llama/Llama-3.2-1B tokenizer, however this model is gated and requires a Hugging Face token in order access it. A Hugging Face access token is a personal API key that is used to securely authenticate programmatic access to Hugging Face services.
- **Configuring CAS Query Access Control and Security**
  Content-Aware Storage (CAS) supports domain-level authorization, Identity Provider integration, and optional file-level security to control query access.

Integrating CAS and IBM Data Cataloging

Rule-based integration between IBM Data Cataloging and Content-Aware Storage (CAS) ensures selective, policy-driven data ingestion into CAS vector stores.

By integrating IBM Data Cataloging with CAS, organizations can apply advanced pre-filtering through configurable, rule-based policies that control which files are ingested into CAS.

Administrators can define criteria such as file type, size, location, and content patterns to ensure that only relevant data is stored in CAS vector stores. Custom tags and attributes further enhance this capability by providing fine-grained control over the ingestion process. For more information, see Managing IBM Data Cataloging rules when integrated with CAS.

- **Integrating IBM Data Cataloging for file filtering**
  You can enable integration of IBM Data Cataloging with Content-Aware Storage (CAS) to configure file ingestion filtering. Any file from a data source that meets the filtering criteria can be ingested into the corresponding domain. Filtering is enforced across all domains within the same IBM Fusion cluster.
- **Using IBM Data Cataloging attributes with CAS**
  Content-Aware Storage (CAS) integrates with IBM Data Cataloging service (DCS) to support automated file categorization through flexible, attribute-based tagging.
- **IBM Data Cataloging and Content-Aware Storage insights**
  IBM Data Cataloging and Content-Aware Storage (CAS) introduce data insights dashboards that provide immediate visibility into data once sources are connected and scanned. This experience is enabled when both IBM Data Cataloging and CAS are integrated. The provided insights surface key indicators of data composition on each established data source.

Integrating IBM Data Cataloging for file filtering

You can enable integration of IBM Data Cataloging with Content-Aware Storage (CAS) to configure file ingestion filtering. Any file from a data source that meets the filtering criteria can be ingested into the corresponding domain. Filtering is enforced across all domains within the same IBM Fusion cluster.

Before you begin

- Ensure that you have IBM Fusion 2.12.0 version.
- Install IBM Data Cataloging 2.3.0. For more information, see Installing IBM Data Cataloging.
- Connect to Scale Remote Filesystem from IBM Data Cataloging by using these instructions: Creating an IBM Storage Scale data source connection. For more information, see IBM Storage Scale data source connections.
- Understand the default filtering criteria. For more information, see Managing IBM Data Cataloging rules when integrated with CAS.

Enable or disable filtering through CLI

Enable filtering

To enable filtering by using IBM Data Cataloging, run the following command:

```
oc patch casinstall ibm-cas-service-instance -n ibm-cas --type=merge -p '{"spec":{"enabled_features":{"dcs":
{"enable":true,"instance_name":"data-cataloging-service-instance","namespace":"ibm-data-cataloging"}}}}'
```

To confirm filtering is enabled, run the following command:

```
DCS_CR=$(oc get SpectrumDiscover -n ibm-data-cataloging -o jsonpath="{.items[*].metadata.name}")
oc get SpectrumDiscover $DCS_CR -n ibm-data-cataloging -o jsonpath='{.status.conditions[*]}{"\n"}'
```

Tip: You must see the type `ContentAwareStorage` listed as `SuccessfullyEnabled`:

```
{"lastTransitionTime":"2025-09-
10T08:33:42Z","message":"","reason":"SuccessfullyEnabled","status":"True","type":"ContentAwareStorage"}
```

Disable filtering

To disable filtering, run the following command:

```
oc patch casinstall ibm-cas-service-instance -n ibm-cas --type=merge -p '{"spec":{"enabled_features":{"dcs":
{"enable":false,"instance_name":"data-cataloging-service-instance","namespace":"ibm-data-cataloging"}}}}'
```

To confirm filtering is disabled, run the following command:

```
DCS_CR=$(oc get SpectrumDiscover -n ibm-data-cataloging -o jsonpath="{.items[*].metadata.name}")
oc get SpectrumDiscover $DCS_CR -n ibm-data-cataloging -o jsonpath='{.status.conditions[*]}{"\n"}'
```

Tip: You must no longer see the type `ContentAwareStorage` listed in the output.

One-click installation and enablement integration experience

CAS provides a one-click installation of IBM Data Cataloging directly from the Content-Aware Storage > Settings page. This integration streamlines the setup process by enabling administrators to install and configure IBM Data Cataloging without leaving the CAS user interface, improving the integration between both data services.

Overview

When IBM Data Cataloging is not installed, the Advanced filtering and scanning section in the Content-Aware Storage > Settings page displays an Install Data Cataloging button. Clicking this button launches the same installation flow that is used in the Services page. As a result, no additional configuration is required outside of CAS.

After installation completes, the same section enables controls to manage IBM Data Cataloging filtering:

- Modify rulesets: Opens the OpenShift® Container Platform custom resource editor to configure filtering rulesets. For more information, see Managing IBM Data Cataloging rules when integrated with CAS.
- Enable / Disable toggle: Enables or disables IBM Data Cataloging-based advanced filtering without uninstalling the service.

Prerequisites

Before starting the installation, ensure that you meet the following prerequisites:

- CAS v1.1.5 is deployed and the Content-Aware Storage > Settings page is accessible.
- The cluster meets the resource requirements for IBM Data Cataloging.

Installing IBM Data Cataloging

1. In the IBM Fusion console, go to Content Aware Storage > Settings.
2. Locate the Advanced filtering and scanning section. If IBM Data Cataloging is not installed, the section displays an Install Data Cataloging button.
3. Click Install Data Cataloging. An installation progress indicator that shows the current percentage replaces the button.
   Installation follows the same setup process as the Services page. If the installation fails, an error message is displayed in the same section with details to help you troubleshoot or check Fusion Events.
4. Wait for the installation to complete. The progress indicator disappears when IBM Data Cataloging is fully installed.

Managing IBM Data Cataloging after installation

After IBM Data Cataloging is installed, the Advanced filtering and scanning section updates to show the management controls.

Control	Description
Modify rulesets	Opens the OpenShift Container Platform custom resource for IBM Data Cataloging, where you can configure the filtering rulesets that determine which content CAS processes.
Enabled / Disabled toggle	Enables or disables IBM Data Cataloging based advanced filtering. Disabling does not uninstall the service.

Modifying rulesets

Click Modify rulesets to open the OpenShift Container Platform custom resource editor. Update the ruleset configuration as needed and save the resource. The changes take effect immediately. For more information, see Managing IBM Data Cataloging rules when integrated with CAS.

Enabling and disabling filtering

Use the toggle in the Advanced filtering and scanning section to enable or disable IBM Data Cataloging filtering at any time. When filtering is disabled, CAS continues to operate without IBM Data Cataloging-based content analysis until filtering is re-enabled.

Using IBM Data Cataloging attributes with CAS

Content-Aware Storage (CAS) integrates with IBM Data Cataloging service (DCS) to support automated file categorization through flexible, attribute-based tagging.

Attributes can be applied in DCS either before or after files are ingested into CAS, enabling administrators to enrich data with meaningful metadata such as document type, department, project, or other custom attributes. After attributes are applied in DCS, you can use the CAS Query Service API to perform filtered queries that return only the most relevant content chunks.

Creating a tag

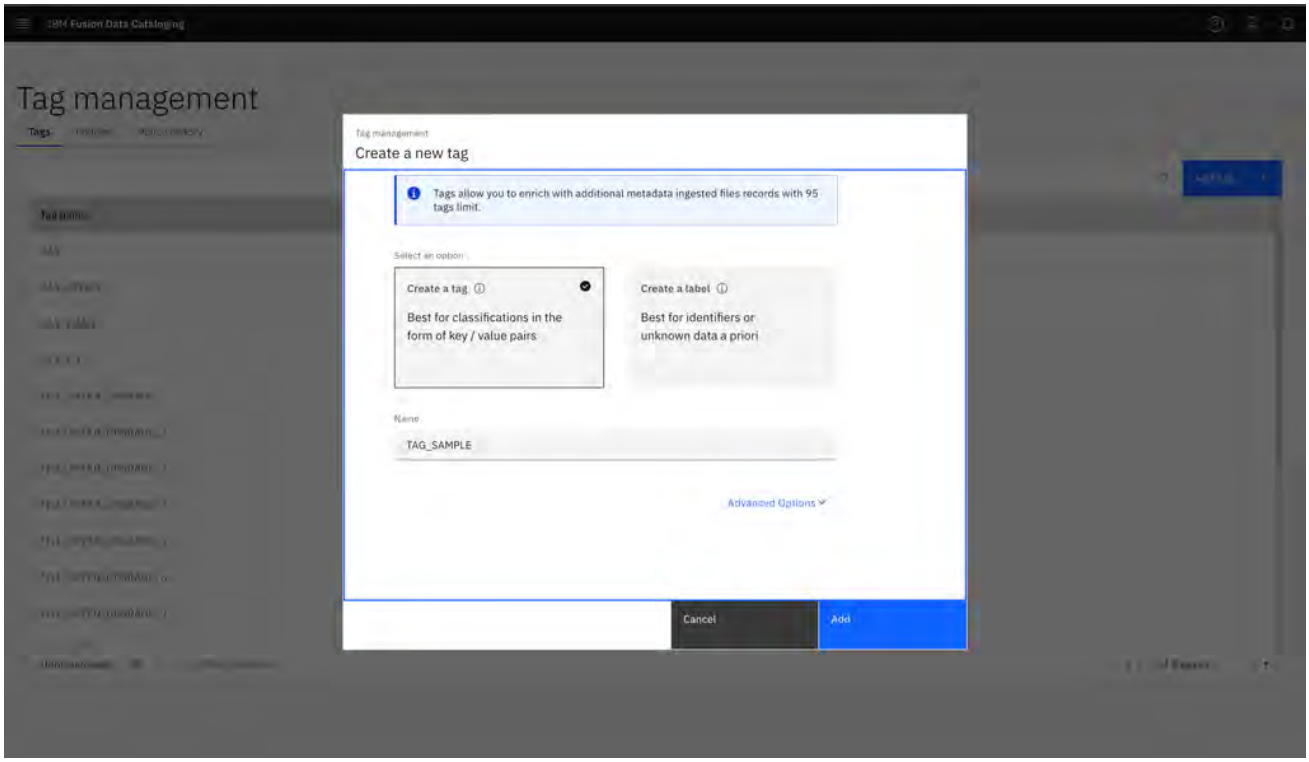
In DCS, additional metadata is applied by creating tags. A tag is a custom metadata field that enhances existing metadata with organization-specific information and supports data classification, search, and reporting.

To create a tag, follow these steps:

1. In the IBM Fusion Data Cataloging UI, open the menu and go to Metadata > Tag management.
2. Open the Tags section, and click Add Tag.



The Create a new tag window opens.



3. In the Name field, specify a tag name, and click Add.

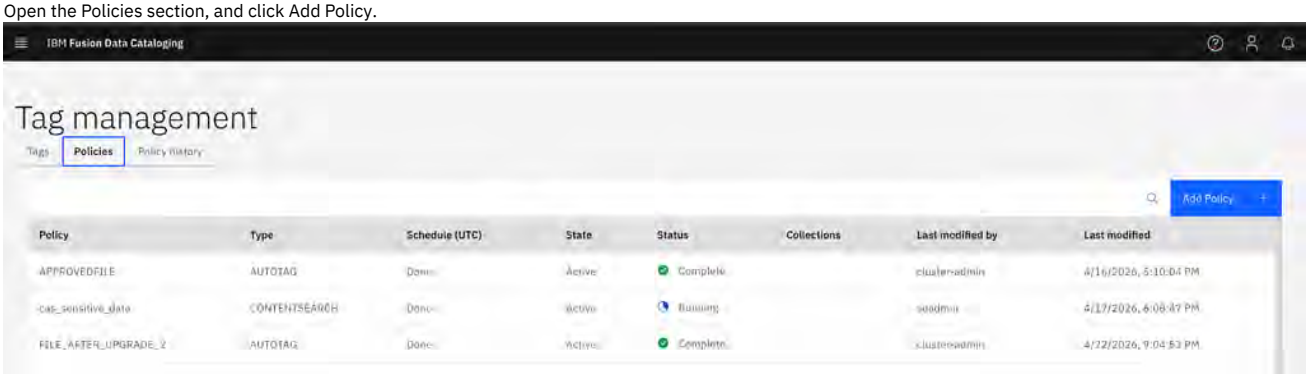
For additional details about tags and how to create them, see Managing tags.

Creating a policy

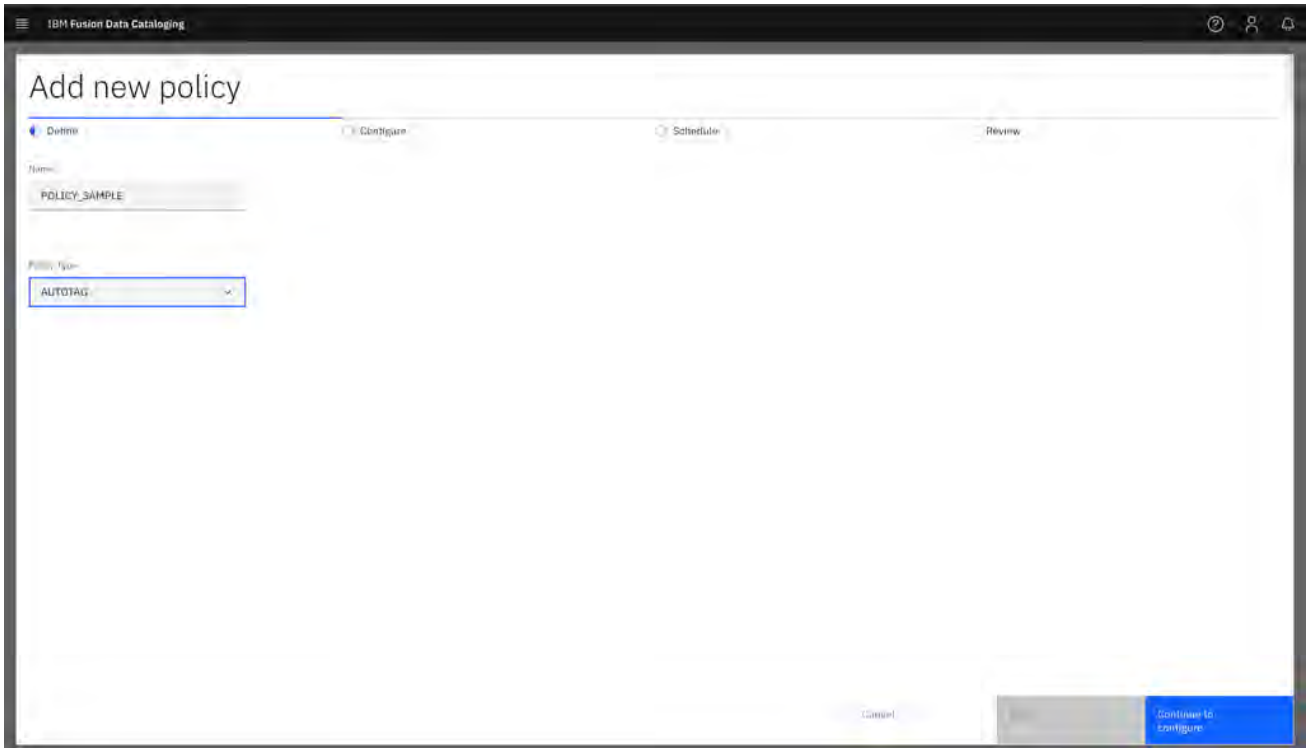
After a tag is created, a DCS policy must be defined to associate documents with that tag. The policy uses a user-defined query to classify documents, and when the policy is executed, the tag is automatically populated with values for all matching documents.

To create a policy, follow these steps:

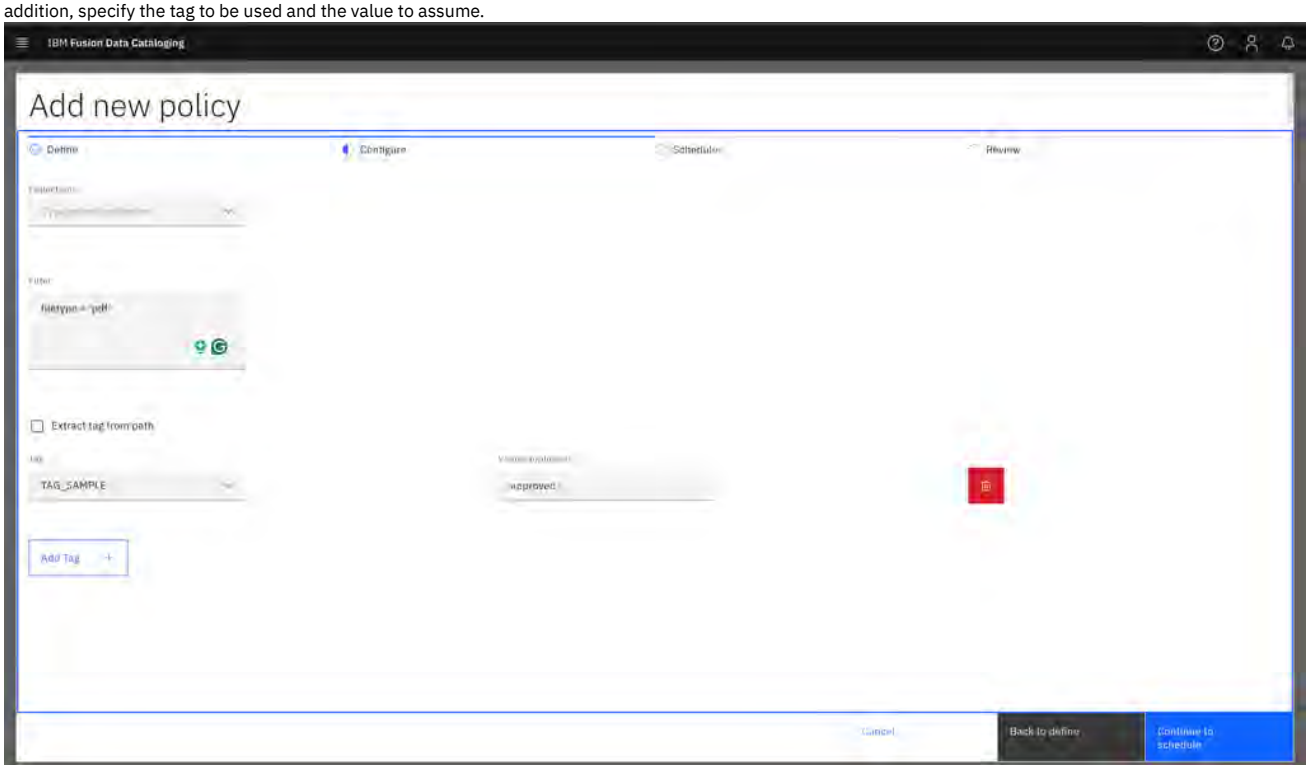
1. In the IBM Fusion Data Cataloging UI, open the menu and go to Metadata > Tag management.
2. Open the Policies section, and click Add Policy.



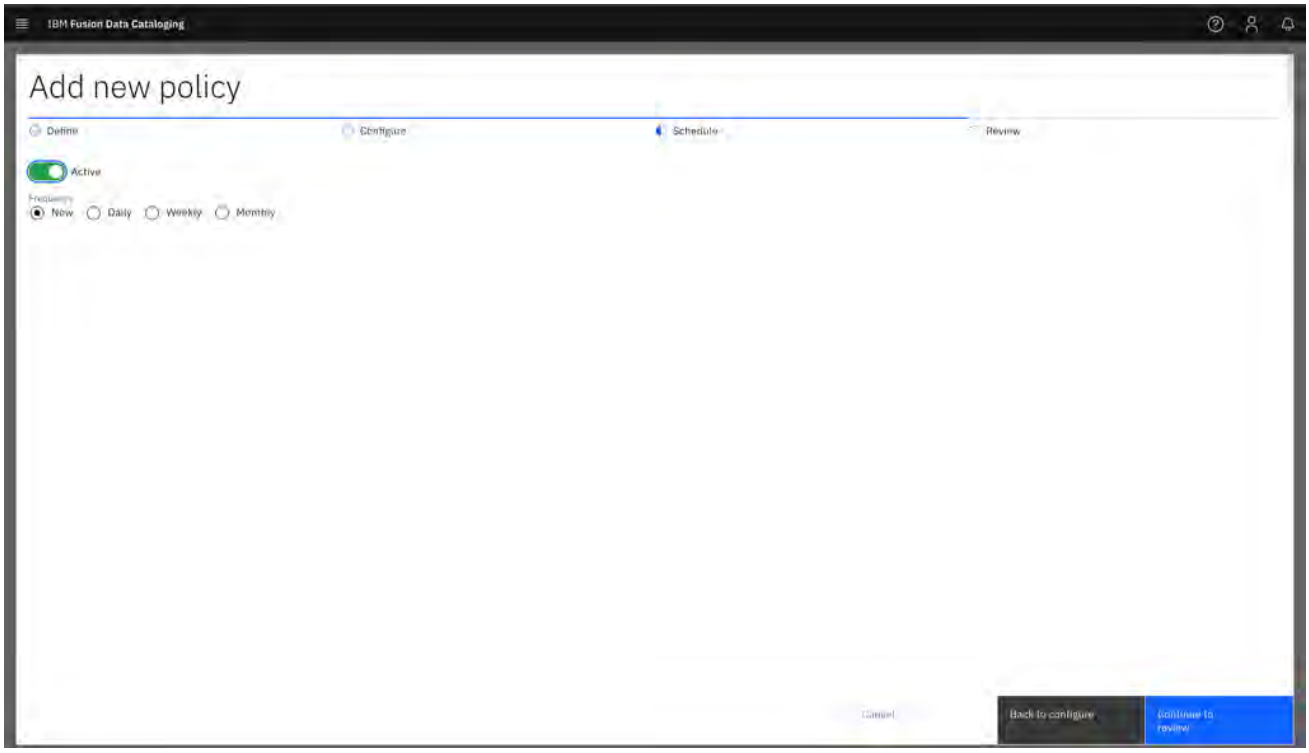
The Add new policy window opens.



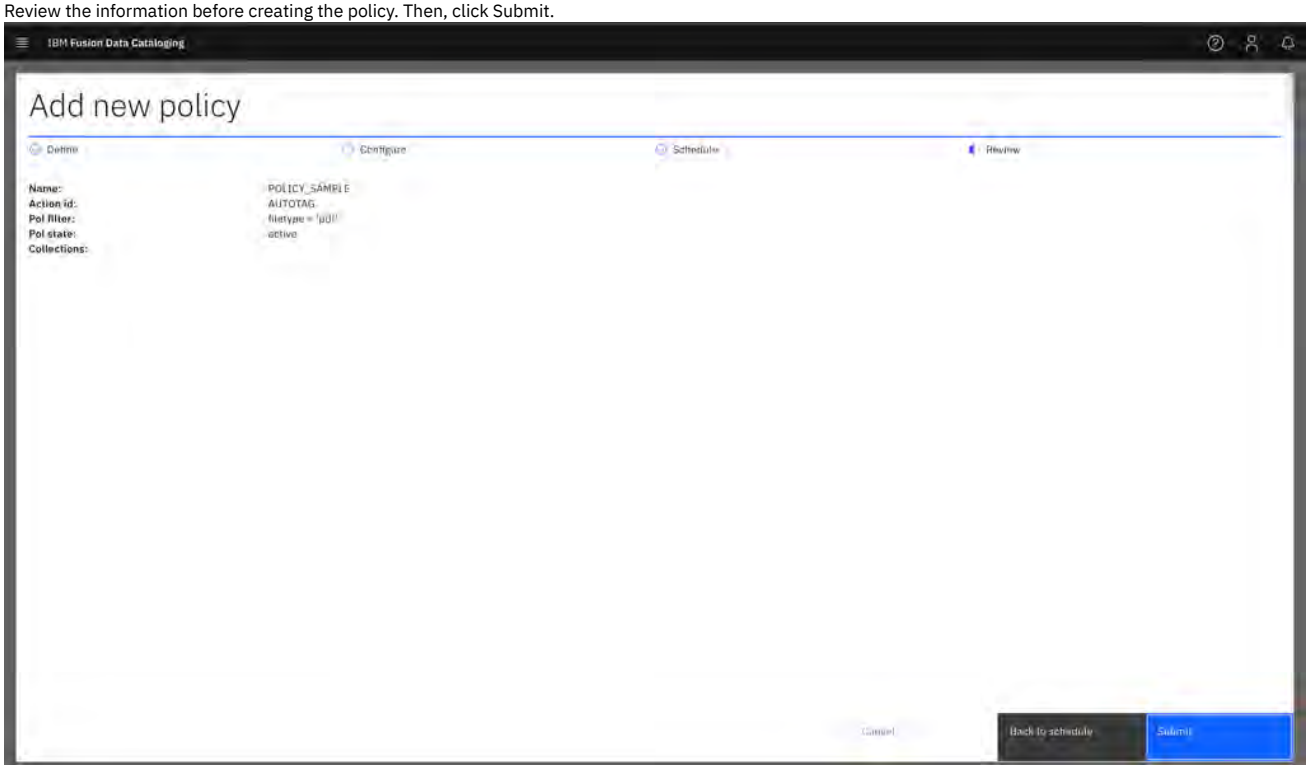
3. In the Name field, specify a name for the new policy.
4. Select the Policy Type. The default value is AUTOTAG policy.
5. Click Continue to configure.
6. Configure the policy. This action includes specifying a filter, which is used by the mechanism to tag files by tagging only those files that match the specified filter. In addition, specify the tag to be used and the value to assume.



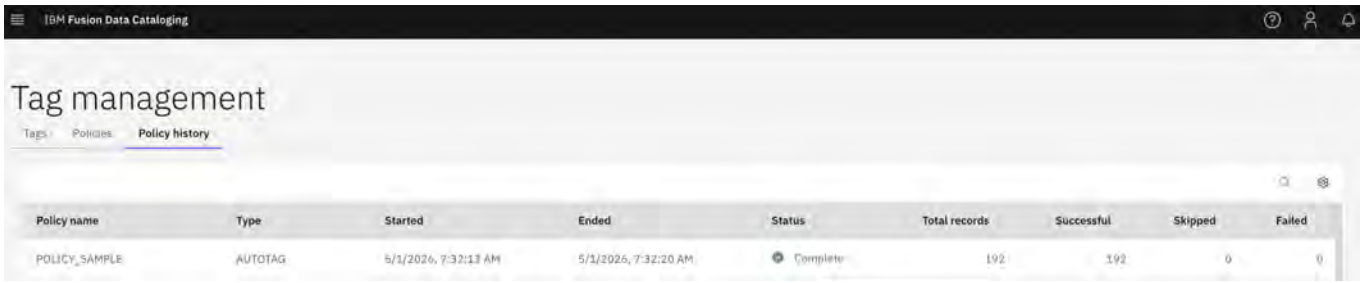
7. Click Continue to schedule.
8. Configure a schedule to execute the policy immediately, or at any specific point in the future. This action automatically executes the policy periodically.



9. Click Continue to review.
10. Review the information before creating the policy. Then, click Submit.



You can monitor the policy's progress by reviewing the Policy history records.



For additional instructions on creating policies, see Adding policies.

IBM Data Cataloging and Content-Aware Storage insights

IBM Data Cataloging and Content-Aware Storage (CAS) introduce data insights dashboards that provide immediate visibility into data once sources are connected and scanned. This experience is enabled when both IBM Data Cataloging and CAS are integrated. The provided insights surface key indicators of data composition on each established data source.

The UI dashboard surfaces foundational indicators of the associated data sources from the storage. No additional configuration is required to start using the dashboard.

Prerequisites

- Content-Aware Storage v1.1.5
- IBM Data Cataloging v2.5.3
- IBM Storage Scale v6.0.0.2 or later
- Ensure that both services are integrated. For more information, see Integrating IBM Data Cataloging for file filtering.

Overview

When a data source is added on CAS, the dashboards display insights that are scoped to that source.

You can go to Fusion UI > Content-Aware Storage > Data Sources > Details View.

The experience is organized into the following sections:

- Data source summary: High-level file and storage metrics.
- Data composition: Breakdown of file types and file size distribution.

Data source summary

At the top of the Data source view, four summary cards display key metrics for the selected data source.

Metric	Description
Total files	Total number of files that are indexed by IBM Data Cataloging across all data sources and prefixes for the data source.
Total storage	Aggregate storage capacity consumed, reported in GiB.
Ready to process	Number and percentage of files that match the CAS ingestion criteria defined in the active ruleset. For more information, see Managing IBM Data Cataloging rules when integrated with CAS. When IBM Data Cataloging and CAS are integrated, this value reflects the result of the rules that are specified in the custom resource.
Sensitive data	Count of files flagged as containing sensitive content. See the Sensitive data section for details on how this count is determined.

Data composition

The Data composition section provides a structural breakdown of the cataloged dataset across two dimensions.

File types and formats
: Files are grouped by category (Documents, Images, Videos, Audio, Other). Each category shows the total file count and storage size, and distinguishes between formats that CAS can process and those it cannot. Expanding a category reveals the individual extensions within each group.

  The following file formats are supported for CAS ingestion:

  `pdf`, **`doc`**, **`docx`**, **`txt`**, **`html`**, **`markdown`**, **`md`**, **`pptx`**, **`jpeg`**, **`png`**, **`bmp`**, **`xlsx`**, **`asciidoc`**, **`adoc`**, **`xhtml`**, **`csv`**, **`webp`**, **`wav`**, **`ogg`**, **`opus`**, **`mp3`**

  For more information, see Managing IBM Data Cataloging rules when integrated with CAS. All other extensions appear under the Not Supported label within their category.

File size distribution
: A horizontal bar chart shows the number of files that are grouped by size range. The buckets used are: < 1 MB, 1-10 MB, 10-100 MB, 100 MB-1 GB, and > 1 GB.

  This metric is derived from a single aggregated API query against the IBM Data Cataloging catalog.

Creating a data source

A data source is a repository or a system that stores and provides data for use in applications or systems. They are storage locations that provide data to use in AI workflows.

Before you begin

- For prerequisites specific to S3 type data source, see S3 type data source.
- For prerequisites and limitations specific to NFS data sources, see Network File System (NFS) type data source.

About this task

- The data source serves as the foundation for feeding information into the data pipeline.
  You have the options of:
  - Object Storage such as Amazon S3
  - IBM Storage Scale
  - Network File System (NFS)
- Content-Aware Storage (CAS) does not directly interface with S3 storage providers. It accesses S3 storage through the IBM Storage Scale S3 filesystem and with Active File Management (AFM) feature enablement provides a cached copy of the S3 content.
- Up to 25 CAS data sources are supported. Each data source can be one of the following:
  - An external S3 bucket using a IBM Storage Scale AFM fileset to ingest the S3 content.
  - A fileset residing on the IBM Storage Scale filesystem attached to CAS (without AFM).
- CAS supports multiple IBM Storage Scale filesystems. However, all remote filesystems must belong to a single remote IBM Storage Scale cluster.
- The following constraints exist for change notifications for filesets that reside in the IBM Storage Scale file system is remote mounted by IBM Fusion Global Data Platform service (without AFM):
  - Up to 10 million files per fileset
  - Up to 100 million files total monitored
- Only independent fileset is supported.

Procedure

1. From the menu, go to Content-aware storage > Data source.
2. In the Data sources page, click Connect data source.
3. Enter the name of the Data source.
4. Select the storage type and click Next.
   The available storage types are IBM Cloud, Storage Scale, AWS, S3 Compliant, and NFS.
5. Enter the following details in the Connection details page.
   The Connection details page vary based on your storage type selection.
   - IBM Cloud, AWS, or S3 Compliant
     - Enter the Endpoint.
       It refers to the URL through which you can access the bucket and its contents. For more information about Endpoint rules, see AWS documentation.
     - Enter the name of the S3 Bucket.
       For more information about the bucket naming guidelines, see AWS documentation.
     - For AWS, enter the Region. It is the Amazon Web Services region where the bucket is located.
     - Enter the Access key and Secret access key
       These are security credentials needed to access the contents of the bucket.
     - In the Certificate settings section, enter the Secret name for certificate.
       This is an optional parameter. SSL secured object storage locations require certificates for authentication. Create an OpenShift TLS secret in the **`ibm-spectrum-fusion-ns`** namespace or your IBM Fusion namespace. Provide the secret name for the credentials. For more information about creating a secret, see Creating a secret.
   - Storage Scale
     Enter the Path. It is the junction path.
   - NFS
     Provide the following information:
     - NFS server: Enter the NFS server hostname or IP address.
       Note: Although this field is defined as a list, only a single NFS server is supported in the CAS 1.1.5 release.
     - Export path: Enter the NFS export path to monitor. For example: /home/user/data
6. In the Caching filesystem section, select the file system where the cache of this data source must be stored.
   If there is only one remote file system detected, it gets automatically selected and this field is not available for selection.
7. Click Connect to submit the information so that CAS can be enabled for the data source.

What to do next

For IBM Storage Scale type data source, determine the group owner that has read and execute permission to the junction path to allow CAS to read its files and view the directory.
Log in to your Scale cluster to check the permission. For example, if your junction path is /gpfs/gpfs3/my-data/, run the following command to change the path:

```
cd /gpfs/gpfs3/
```

```
ls -la
```

Example output:

```
drwxr-x---  2 root cas    4096 May 22 19:50 my-data
[root@tc11scale1 gpfs3]# getent group cas | cut -d: -f3
310
```

If the group owner is not root, then run the following command to add the annotation to the previously created datasource. In this example, the group owner is **`cas`**.

```
oc annotate DataSource datasource-name group-id='310' --overwrite
```

Here, the **`datasource-name`** and **`310`** are example values that can change depending on the datasource name and the GID set in Scale.

- **Prerequisites for S3 and NFS type data sources**
  Prerequisites for creating S3 and NFS type data sources in CAS, including Active File Management requirements, version constraints, and AWS permissions.
- **Deploying an S3 data source with custom SSL certificate**
  Deploy an S3 data source with a custom SSL certificate on IBM Storage Scale 6.0.0.x by manually creating the AFM fileset before connecting the data source.

Prerequisites for S3 and NFS type data sources

Prerequisites for creating S3 and NFS type data sources in CAS, including Active File Management requirements, version constraints, and AWS permissions.

S3 type data source

Prerequisites

Ensure that you meet the following prerequisites if you plan to use an S3 type data source:

- If you choose a remote filesystem as the cache filesystem for your S3 type data source,
  - Enable Active File Management (AFM) on the remote IBM Storage Scale cluster. For more information, see the Scale documentation.
  - The remote IBM Storage Scale cluster must be at least at version 5.2.3 and the IBM Fusion cluster must be at least at IBM Fusion level 2.10.
- Starting with IBM Storage Scale 6.0.1, the Amazon Web Services account used to provide the access key and secret key for bucket authentication must be granted both `AmazonSNSFullAccess` and `AmazonSQSFullAccess` permissions. For more information, see the Scale AFM documentation.

Network File System (NFS) type data source

Prerequisites

Ensure that you meet the following prerequisites if you plan to use an NFS type data source:

- If you choose a remote filesystem as the cache filesystem for your NFS type data source,
  - Enable Active File Management (AFM) on the remote IBM Storage Scale cluster. For more information, see the Scale documentation.
  - The remote IBM Storage Scale cluster must be at least at version 5.2.3.
- The IBM Fusion cluster must be at least at IBM Fusion level 2.13 and Fusion Data Foundation must be at 4.21.x.

Limitations and considerations

The following limitations apply to NFS data sources:

- In this CAS release 1.1.5, the integration with IBM Data Cataloging is not supported.
- Only a single NFS server per NFS export path is supported.
- Only new NFS filesets that are created through CAS are supported. NFS data source does not support existing NFS filesets that are previously created in IBM Storage Scale.
- Kerberos-enabled NFS exports are not supported.
- File-level access control lists (ACLs) are not supported. Only domain-level ACLs are applicable.
- The NFS server and export path are fixed at data source creation time and cannot be updated later due to AFM limitations.

Deploying an S3 data source with custom SSL certificate

Deploy an S3 data source with a custom SSL certificate on IBM Storage Scale 6.0.0.x by manually creating the AFM fileset before connecting the data source.

Important: The following procedure applies only to IBM Storage Scale version 6.0.0.x.
For IBM Storage Scale 6.0.1.x and later, custom Transport Layer Security (TLS) certificates are configured automatically when you create a data source. Specify the certificate secret name in the Certificate settings section. Certificate rotation occurs automatically when the secret is updated. Follow the standard procedure described in Creating a data source.

Procedure (IBM Storage Scale 6.0.0.x only)

1. For Filesystem with IBM Storage Scale version 6.0.0.x, create a data source for an S3 bucket, which has a custom Secure Sockets Layer (SSL) certificate by adding bucket name, end-point, secret key, and access key details as mentioned in Creating a data source.
   The data source status is updated to `Authentication Failure` as the SSL cannot be processed.
2. Log in to the IBM Storage Scale 6.0.0.x cluster.
3. Get the AFM cache fileset name by listing the configuration details on the filesystem with the following command, and identify the one for the S3 bucket you just tried to add:

   ```
   mmafmcoskeys all get --report
   ```

4. Retrieve the `fileset-name` from the export map, which is in the following format:

   ```
   <bucket-name>:<fileset-name>-exportmap=<keys>
   ```

5. Copy the TLS certificate on a path, for example at /var/cas/tls-cert/sample-cert.pem.
6. Run the following command to re-create the fileset and make it `Active`:

   ```
   mmafmcosconfig <fs> <fileset-name> --endpoint 'https://<fileset-name>-exportmap:443' --object-fs --bucket <bucket-name> --tmpdir '.cachevolumetmp' --gid <gid> --mode 'iw' --ssl-cert-verify --ssl-cert-path <cert-with-full-path> --async-prefetch-interval 60 --makeactive
   ```

   The fileset must be deployed with the following output:

   ```
   Partition <fileset-name> is created and policy broadcasted to all nodes.
   ```

7. Check the status of the fileset:

   ```
   /usr/lpp/mmfs/bin/mmafmctl <fs> getstate -j <fileset-name>
   ```

   Remember: The fileset must be in the `Active` state and the contents of S3 bucket must be readable in the fileset.
8. Delete the previously failed data source and redeploy it with the same bucket information.

Results

The data source gets **Connected** as it can be able to use the manually created fileset.

Creating a domain and connecting with data source

Domains collect, transform, and transfer data from multiple data sources to a destination, enabling efficient data flow for analysis and AI models.

About this task

When you configure a domain, you automatically configure the **DocumentProcessorCR**. The processor streamlines data flow for analysis, reporting, and powering AI models effectively.

Procedure

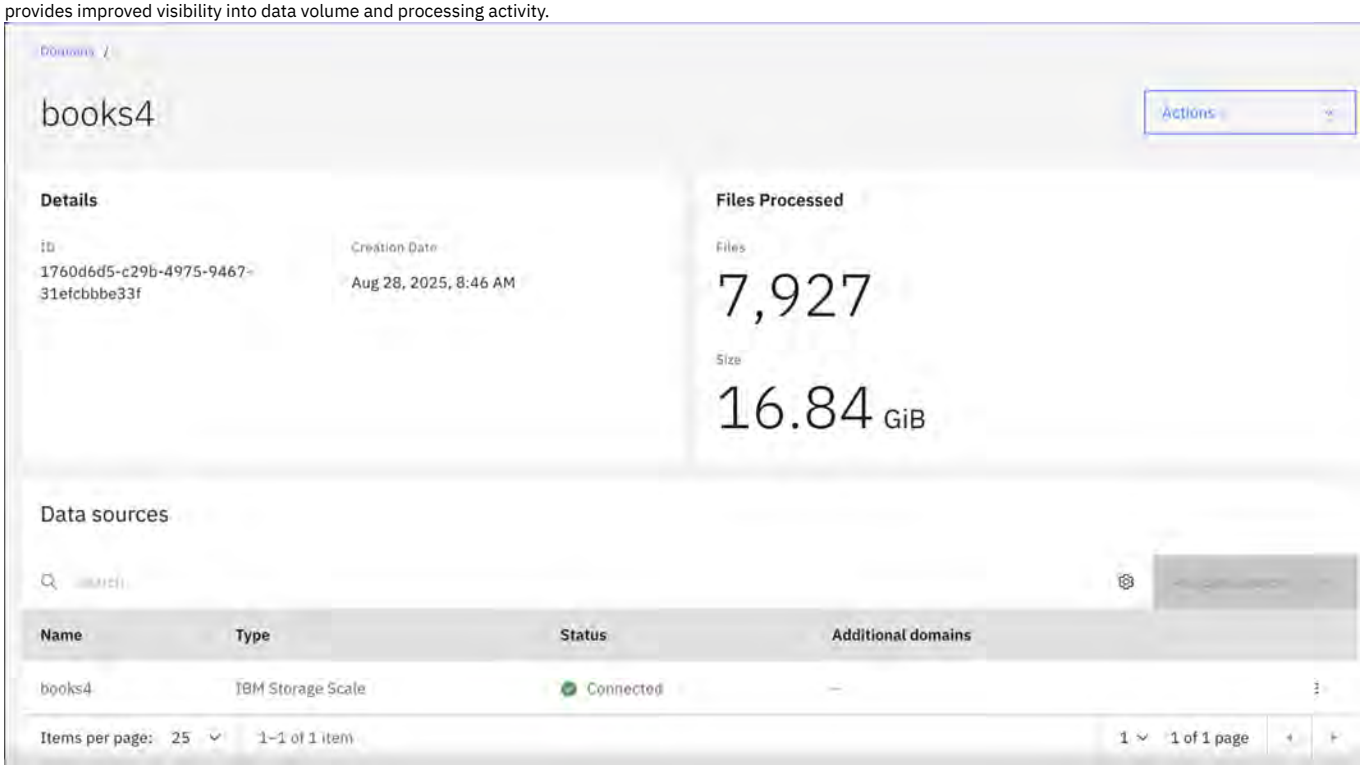
1. Go to Content-aware Storage > Domains.
2. Click Create domain.
3. In the Create domain dialog, enter the name of the domain and select the document processor engine.
   Note: NVIDIA multimodal requires NVIDIA NIM to be configured first.
4. Click Next.
5. Select the content types to be processed by the domain (only applicable to NVIDIA multimodal). Then click Next.
6. Select an option to either connect a new data source or use an existing one.
   a. If you choose to use an existing data source, click Finish. The domain gets created.
   b. If you choose to connect a new data source, click Next and proceed with the next steps to enter the required data source details. For more information about the data source fields, see Creating a data source. Upon completing, both the data source and domain get created.

   If the status of the data source gets stuck in connecting state, ensure that you enable watch creation. For more information about the configuration, see Configuring Scale user to enable watch creation.

   Note: You can delete a domain from the Domain details page only if no data source exists for it.

What to do next

You can view file processing metrics and data source status for each domain. Metrics include the total number of files and the file size within the selected domain that provides improved visibility into data volume and processing activity.



Tip: To customize the metrics polling interval, set the **METRICS_POLLING_INTERVAL_MS** parameter in the **cas-config** ConfigMap located in the CAS installation namespace:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: cas-config
  namespace: ibm-cas
data:
  ENABLE_FILE_LEVEL_SECURITY: 'true'
  HF_ACCESS_TOKEN_SECRET_NAME: hf-access-token
```

```
METRICS_POLLING_INTERVAL_MS: '60000' ---> Metrics Polling interval in millisecond
NVMM_EMBED_SERVICE: 'http://nv-ingest-embedqa.nv-ingest.svc.cluster.local'
NVMM_NIM_SERVICE: 'http://nv-ingest.nv-ingest.svc.cluster.local'
```

Configuring Document Processor Engine

Steps to configure Document Processor Engines: IBM Docling Multimodal and NVIDIA Multimodal.

- **Supported document types**
  List of supported document types for IBM Docling Multimodal and NVIDIA Multimodal document processing engines.
- **IBM Docling Multimodal processing engine**
  IBM Docling Multimodal is an open source, document processing engine that is designed to convert unstructured documents into structured, machine-readable formats for use in AI applications. It enables you to extract, parse, embed, and store document vectors by using Docling with Granite embedding models.
- **NVIDIA Multimodal processing engine settings**
  This section enables you to configure the NVIDIA Multimodal pipeline.

Supported document types

List of supported document types for IBM Docling Multimodal and NVIDIA Multimodal document processing engines.

IBM Docling Multimodal

The supported document types for text-based extraction include:

- `.pdf`
- `.txt`
- `.md`
- `.asciidoc`
- `.csv`
- `.html`
- `.xhtml`
- `.docx`
- `.pptx`
- `.xlsx`

As of now, other file types without text are not supported (for example, image and audio files).

NVIDIA Multimodal

The supported document types for NeMo Retriever Library include:

- `.docx`
- `.html`
- `.json`
- `.md`
- `.pdf`
- `.pptx`
- `.sh`
- `.txt`

The following image file types are supported with Image Extraction enabled with the `nvidia_multimodal` pipeline. This configuration requires the image captioning NIM to be deployed and configured:

- `.bmp`
- `.tiff`
- `.png`
- `.jpeg`

The following audio file types are supported with the `nvidia_multimodal` pipeline. This configuration requires the audio NIM to be deployed and configured:

- `.mp3`
- `.wav`

IBM Docling Multimodal processing engine

IBM Docling Multimodal is an open source, document processing engine that is designed to convert unstructured documents into structured, machine-readable formats for use in AI applications. It enables you to extract, parse, embed, and store document vectors by using Docling with Granite embedding models.

Enabling Docling Multimodal document processor engine

To enable the Docling Multimodal document processor engine through command line, access the `CasInstall` Custom Resource (CR) and add the following configuration under the spec section:

```
documentProcessingEngine:
  docling_multimodal:
    enabled: true
```

Alternatively, the `CasInstall` CR instance can be patched to enable `docling_multimodal`, as shown in the following example:

```
oc patch casinstalls.cas.isf.ibm.com ibm-cas-service-instance -n ibm-cas --type=merge -p '{"spec":{"documentProcessingEngine":
{"docling_multimodal":{"enabled":true}}}}'
```

Disabling Docling Multimodal document processor engine

Disabling the Docling Multimodal document processor engine removes the Docling services and no longer allows ingestion or query to any domains, which are configured to use Docling Multimodal.

To disable the Docling Multimodal document processor engine through command line, access the `CasInstall` CR and add the following configuration under the spec section:

```
documentProcessingEngine:
  docling_multimodal:
    enabled: false
```

Alternatively, the `CasInstall` CR instance can be patched to disable `docling_multimodal`, as shown in the following example:

```
oc patch casinstalls.cas.isf.ibm.com ibm-cas-service-instance -n ibm-cas --type=merge -p '{"spec":{"documentProcessingEngine":
{"docling_multimodal":{"enabled":false}}}}'
```

- **Configuring Docling Multimodal GPUs**
  This section describes how to configure Docling Multimodal GPUs.

Configuring Docling Multimodal GPUs

This section describes how to configure Docling Multimodal GPUs.

The default GPU resource that is used by the Docling services is `nvidia.com/gpu`. If the GPUs are configured to use Multi-Instance GPU (MIG), then a corresponding MIG profile must be assigned to each service for it to start properly. The IBM Docling Multimodal resources can be overridden in the `CasInstall` Custom Resource (CR). Each deployment that requires a GPU has an override flag for use with alternative resource configurations.

To alter the Docling Multimodal GPU targets from the command line, access the `CasInstall` CR and add the following flags under the `spec` section:

```
spec:
  flags:
    - VLLM_VISION_GPU=<gpu target>
    - VLLM_EMBEDDING_GPU=<gpu target>
    - DOCLING_GPU=<gpu target>
```

For example, to specify an alternative Multi-Instance GPU (MIG) indicator for all, add the following flags under the `spec` section:

```
spec:
  flags:
    - VLLM_VISION_GPU=nvidia.com/mig-7g.80gb
    - VLLM_EMBEDDING_GPU=nvidia.com/mig-7g.80gb
    - DOCLING_GPU=nvidia.com/mig-7g.80gb
```

Result: The GPU resource indicator is then automatically detected and a new deployment is rolled out.

NVIDIA Multimodal processing engine settings

This section enables you to configure the NVIDIA Multimodal pipeline.

The `DocumentProcessor` custom resource (CR) supports the following configuration options for controlling the NVIDIA Multimodal file processing engine. Each option affects content chunking, processing performance, and search quality. When you select a processing approach, consider the tradeoffs among these factors.

NVIDIA Multimodal `DocumentProcessor` CR configuration options and considerations

Option	Valid option values	Description	Considerations
`spec` > `tasks` > `name` > `nvidiaExtract` > `extractText`	`True`/`False` (Default: `True`)	Extracts the actual text from a document, including the text in tables, charts, and infographics.	Extract text is the most performant for ingestion.
`spec` > `tasks` > `name` > `nvidiaExtract` > `extractTables`	`True`/`False` (Default: `True`)	Extracts text by summarizing each table.	A second chunk is created if `extractText` is also selected.
`spec` > `tasks` > `name` > `nvidiaExtract` > `extractCharts`	`True`/`False` (Default: `True`)	Extracts text by summarizing each chart.	A second chunk is created if `extractText` is also selected.
`spec` > `tasks` > `name` > `nvidiaExtract` > `extract` > `infographics`	`True`/`False` (Default: `True`)	Extracts text by summarizing each infographic.	A second chunk is created if `extractText` is also selected.
`spec` > `tasks` > `name` > `nvidiaExtract` > `extractImages`	`True`/`False` (Default: `True`)	Extracts caption by summarizing each image.	Impacts the time of ingestion.

Option	Valid option values	Description	Considerations
`spec` > `tasks` > `name` > `nvidiaExtract` > `tableOutputFormat`	Enumeration: `markdown`, `pseudo_markdown`, `simple` (Default: `pseudo_markdown`)	Extracted tables are returned in a particular format to be returned as the text that is vectorized.	N/A
`spec` > `tasks` > `name` > `nvidiaExtract` > `extractTextDepth`	Enumeration: `page`, `document` (Default: `page`)	Extracts text at levels of granularity.	Impacts the number of chunks that are stored and the size of text returned.
`spec` > `tasks` > `name` > `nvidiaSplit` > `chunkSize`	Numeric (Default: `1024`)	Size of each chunk in number of bytes.	For more information on all splitting options, see Split Documents.
`spec` > `tasks` > `name` > `nvidiaSplit` > `chunkOverlap`	Numeric (Default: `150`)	Number of bytes to overlap between consecutive chunks.	
`spec` > `tasks` > `name` > `nvidiaSplit` > `tokenizer`	String (Default: `meta-llama/Llama-3.2-1B`)	Identifies the embedding model to split the content with.	
`spec` > `tasks` > `name` > `nvidiaSplit` > `splitSourceTypes`	Array of Strings (Default: `empty`)	Identifies the file types to apply the prior `nvidiaSplit` configuration.	

During the Creating a domain and connecting with data source procedure, the CAS user interface creates a `Domain` custom resource (CR) and an associated NVIDIA Multimodal `DocumentProcessor` CR. The `DocumentProcessor` CR configures the processing engine to use the default settings, which include extracting text only and processing content on a per-page basis. To apply more configuration options, you must create or modify the NVIDIA Multimodal `DocumentProcessor` CR by using the OpenShift console or the CLI.

Configuring NVIDIA Multimodal pipeline

After you create a `Domain` CR in the CAS user interface, a corresponding `DocumentProcessor` CR is generated for that domain. Update the `spec` section of the `DocumentProcessor` CR with the required configuration options.
Note: Changes to processing options affect only newly ingested or updated documents. Previously processed documents are reingested with the new settings only when the original document is modified.

1. Find the `DocumentProcessor` CR that is associated with the `Domain`.
   Tip: The `DocumentProcessor` CR has the same name as the `Domain`.
   Example:
   - `oc get domains.cas.isf.ibm.com -n ibm-cas`
   - `oc get documentprocessors.cas.isf.ibm.com -n ibm-cas`
2. View the current configuration of the `DocumentProcessor` CR.
   Example:

   ```
   oc get documentprocessors.cas.isf.ibm.com mydomain -n ibm-cas -o yaml
   ```
3. Ensure that the type is `nvidia_multimodal`. Example: `type: nvidia_multimodal`
4. Modify the `DocumentProcessor` CR according to the options added in the table.
5. Save the `DocumentProcessor` CR.

Results

The CAS operator automatically reconciles the `Deployment` associated with the `DocumentProcessor` CR and the corresponding processing engine pods are restarted with new settings.

Configuring Content-Aware Storage (CAS) through CLI

You can use the CRs to configure CAS.

Ensure all the prerequisites are met. See Planning and prerequisites. To install CAS from the user interface of IBM Fusion, see Installing Content-Aware Storage (CAS).

Create a Scale datasource

```
oc apply -f - <<EOF
apiVersion: cas.isf.ibm.com/v1beta1
kind: DataSource
metadata:
  name: <datasource name>
spec:
 scale:
  fileSystemName: <filesystem name>
  path: <fileset path>
EOF
```

Create a S3 datasource

For prerequisites, create a key/value secret from the OpenShift® console in CAS namespace. For other prerequisites, see Prerequisites for S3 and NFS type data sources.

The following example is based on Amazon AWS.

```
apiVersion: cas.isf.ibm.com/v1beta1
kind: DataSource
metadata:
  name: <datasource name>
  namespace: ibm-cas
spec:
 s3:
  bucket: <bucket name>
  endpoint: <endpoint>
  credentials:
    accessKeyRef:
      key: ""
    secretKeyRef:
      key: ""
    secretName: <Secret name>
  fileSystemName: <filesystem name>
  provider: aws
```

If a Group ID is defined for the underlying fileset, run the following command to add the annotation to the previously created datasource previously:

```
oc annotate DataSource <datasource name> group-id='310' --overwrite
```

Where, `<datasource name>` and `310` are example values that can change depending on the datasource name and the GID set in Scale.

Create a NFS data source

For prerequisites, see Prerequisites for S3 and NFS type data sources.

```
apiVersion: cas.isf.ibm.com/v1beta1
kind: DataSource
metadata:
  name: <datasource name>
  namespace: ibm-cas
spec:
  nfs:
    servers:
      - <server>
    path: "<path>"
    fileSystemName: <filesystem name>
```

Create a domain and add datasource to it

```
kubectl apply -f - <<EOF
apiVersion: cas.isf.ibm.com/v1beta1
kind: Domain
metadata:
  name: <domain name>
spec:
  dataSources:
  - <datasource name>
  collections:
  - name: "<domain name>"
EOF
```

Create a document processor

Document processor **`type`** field refers to the document processing engine that is used to ingest documents. CAS can be configured to use either NVIDIA Multimodal Services or Docling Multimodal Services.

The keys for each type are as follows:

- NVIDIA Multimodal Processing Engine - **`nvidia_multimodal`**

  ```
  oc apply -f - <<EOF
  apiVersion: cas.isf.ibm.com/v1beta1
  kind: DocumentProcessor
  metadata:
    name: <domain name>
  spec:
    type: nvidia_multimodal
    domains:
    - <domain name>
  EOF
  ```

- Docling Multimodal Processing Engine - **`docling_multimodal`**

  ```
  oc apply -f - <<EOF
  apiVersion: cas.isf.ibm.com/v1beta1
  kind: DocumentProcessor
  metadata:
    name: <domain name>
  spec:
    type: docling_multimodal
    domains:
    - <domain name>
  EOF
  ```

Advanced configuration for the NVIDIA Multimodal document processor

To extract content from tables, charts, and images, from the documents in the dataset, enable additional settings in the document processor custom resource (CR).
Note: Enabling these options increases document processing time.
Supported file types for NVIDIA Multimodal document processor: `.pdf`, `.txt`, `.html`, `.json`, `.md`, `.docx`, `.pptx`, `.bmp`, `.jpeg`, `.png`, `.tiff`, `.sh`
Supported options for `nvidia_multimodal` extract task:

- `extractCharts`: Extracts chart graphs from documents.
- `extractTables`: Extracts tabular data from documents.
- `extractImages`: Extracts content from image files and images that are contained within documents.

The following example shows how to configure these options in the `nvidia_multimodal` extract task.

```
oc apply -f - <<EOF
apiVersion: cas.isf.ibm.com/v1beta1
kind: DocumentProcessor
metadata:
  name: <domain name>
spec:
  type: nvidia_multimodal
  tasks:
    - name: Extract
      nvidiaExtract:
        extractCharts: true
        extractImages: true
        extractTables: true
        extractText: true
        extractTextDepth: page
    - name: Split
      nvidiaSplit:
        chunkOverlap: 150
        chunkSize: 1024
        tokenizer: meta-llama/Llama-3.2-1B
    - name: Embed
      nvidiaEmbed: {}
  domains:
  - <domain name>
EOF
```

If you want to configure file ingestion filtering by using Data Cataloging, add the additional supported file types for NVIDIA Multimodal document processor to Data Cataloging's filtering rule:

```
oc patch rule allowed-filetypes -n ibm-data-cataloging --type merge -p '{"spec": {"metadata": [{"field":
"filetype","operator": "in","value": ["doc", "docx", "pdf", "txt", "gif", "html", "json", "md", "pptx", "bmp", "jpeg",
"png", "tiff", "sh" ]}]}}'
rule.datacataloging.ibm.com/allowed-filetypes patched
```

Note: You must update the filtering rule every time you enable Data Cataloging filtering, if you later disable it.

Create a CAS Resource Access Control (CRAC) CR

```
apiVersion: cas.isf.ibm.com/v1beta1
kind: CasResourceAccessControl
metadata:
  name: <CR name: "<Domain name>-access">
  namespace: <CAS installation namespace>
spec:
  #Determines whether file level permission checking can be done for a given domain
  fileACL: <true/false>
  resourceRef:
    name: <Domain name>
    type: Domain
  subjects:
    groups:
      - name: <Name of the IDP/OpenShift group who needs access of given domain>
        role: READER
      # Special group for granting access to all authenticated users
      - name: cas-all-authenticated
        role: READER
    users:
      - name: <Name of the IDP/OpenShift user who needs access of given domain>
        role: READER
```

Note: The `cas-all-authenticated` special group grants domain access to any user who has a valid OpenShift or identity provider (IDP) authentication token. When this group is configured, the system bypasses individual user and group access control (ACL) checks for the domain. This capability is useful for environments where simplified access control is required.
The CAS Resource Access Control (CRAC) CR status contains validation details for resources, users, and groups.

- If resource is not valid, the condition status indicates the exact error.
- If users in the specific user list are not valid, the condition status indicates the name of invalid users.
- If groups in the specific group list are not valid, the condition status indicates the name of invalid groups.
- If the resource, users, and groups are valid, the condition status reflects the validation status message.
  The following example shows how the status message appears in the CRAC CR:

```
status:
  conditions:
    - lastTransitionTime: '2025-08-29T02:15:42Z'
      message: 'Resource: gt20 is validated'
      observedGeneration: 1
```

```
      reason: AsExpected
      status: 'True'
      type: ValidatedResource
    - lastTransitionTime: '2025-08-29T02:15:42Z'
      message: Users are validated
      observedGeneration: 1
      reason: AsExpected
      status: 'True'
      type: ValidatedUsers
    - lastTransitionTime: '2025-08-29T02:15:42Z'
      message: Groups are validated
      observedGeneration: 1
      reason: AsExpected
      status: 'True'
      type: ValidatedGroups
```

To configure CAS to restrict specific users to specific CAS domains through user interface, see Configuring CAS Resource Access Control (CRAC).

To remove search access for a domain through CLI, delete the relevant users and groups from the CRAC CR and save the changes. Removed users and groups then lose authorization to access the query search API for the domain.

Configuring network policies

CAS automatically deploys and manages Kubernetes network policies that enforce a zero-trust communication model across its internal services.

To disable the network policies:

```
oc patch casinstall <install name> -n <cas namespace> \
  --type merge -p '{"spec":{"networkPolicy":{"enabled":false}}}'
```

To enable the network policies:

```
oc patch casinstall <install name> -n <cas namespace> \
  --type merge -p '{"spec":{"networkPolicy":{"enabled":true}}}'
```

Note: The changes are applied automatically, and a restart is not required. You can disable network policies if you want to apply custom policies.

Using your own Hugging Face access token with CAS for NVIDIA Multimodal

NVIDIA NeMo Retriever Library uses models that are downloaded from Hugging Face to do token-based splitting on the content extracted from documents. NVIDIA recommends to use the meta-llama/Llama-3.2-1B tokenizer, however this model is gated and requires a Hugging Face token in order access it. A Hugging Face access token is a personal API key that is used to securely authenticate programmatic access to Hugging Face services.

Before you begin

You must have a valid token to access Meta's Llama 3.2 language models and evals gated repository.

About this task

In Content-Aware Storage (CAS), a Hugging Face access token is used to securely interact with Hugging Face's APIs and services that support intelligent content processing.

For more information about Hugging Face access tokens, see Hugging Face User Access Tokens.

To provide your Hugging Face access token for the meta-llama/Llama-3.2-1B model, perform the following steps:

Procedure

1. Encode the generated Hugging Face access token in Base64 format by using the following command. The resulting value is refereed to as the HUGGING_FACE_ACCESS_TOKEN in the next step.

   ```
   echo "hf_xxxxxxxxxxx" | base64
   ```

2. Create a secret in the CAS installation namespace:

   ```
   kind: Secret
   apiVersion: v1
   metadata:
     name: hf-access-token
     namespace: ibm-cas
   data:
     hf_access_token: HUGGING_FACE_ACCESS_TOKEN
   type: Opaque
   ```

3. Use the encoded access token and update the `cas-install` CR 'spec' section:

   ```
   spec:
     flags:
       - HF_ACCESS_TOKEN_SECRET_NAME=hf-access-token
   ```

Configuring CAS Query Access Control and Security

Content-Aware Storage (CAS) supports domain-level authorization, Identity Provider integration, and optional file-level security to control query access.

You can configure Content-Aware Storage (CAS) query access control and security to suit your requirements and environment by selecting one of the following methods:

- Authorize CAS query users to CAS domains:
  - By default, all users of the Red Hat OpenShift cluster where CAS is deployed can issue CAS queries, depending on the domains they are authorized to access. In this context, they are referred to as CAS query users.
  - Configure CAS to authorize users to specific CAS domains. If an unauthorized CAS query user of a domain runs CAS queries, the query API fails with status code 403 (Invalid Access). For more information, see Configuring CAS Resource Access Control (CRAC).
- Manage CAS users with your own Identity Provider (IDP):
  - Configure CAS to connect to your own IDP. Without this configuration, CAS serves only the Red Hat OpenShift users. For more information, see Configuring CAS to connect to your Identity Provider (IDP).
- Limit certain users to certain files:
  - To enforce stricter access control, you can configure CAS to serve only the content from files for which users have explicit file-level read permissions. Without this configuration, CAS returns results from all ingested data within a domain to any user who has domain-level access, regardless of file-level read permissions. For more information, see Configuring file-level access control in CAS.

Security type	Red Hat OpenShift users	External IDP users
Domain-level access control (mandatory)	Supported	Supported
File-level access control (optional)	Unsupported	Supported; an external IDP is required to federate user information from the Directory server used by the file system.

  - CAS query user setup: To designate users of Red Hat OpenShift or external IDP or Directory server as CAS query users, you must either configure the users or their groups through CAS Resource Access Control (CRAC). For more information, see Configuring CAS Resource Access Control (CRAC).

- **Configuring CAS Resource Access Control (CRAC)**
  Configure users and groups that can access specific Content-Aware Storage (CAS) domains by using the CRAC custom resource.
- **Configuring CAS to connect to your Identity Provider (IDP)**
  An Identity Provider is a system or service that creates, maintains, and manages identity information for users and provides authentication services to applications or other systems. It acts as a trusted third party that authenticates users and shares their identity information with other services.
- **Configuring file-level access control in CAS**
  File-level security in Content-Aware Storage (CAS) ensures that users can access only the data they are authorized to read from the file system.

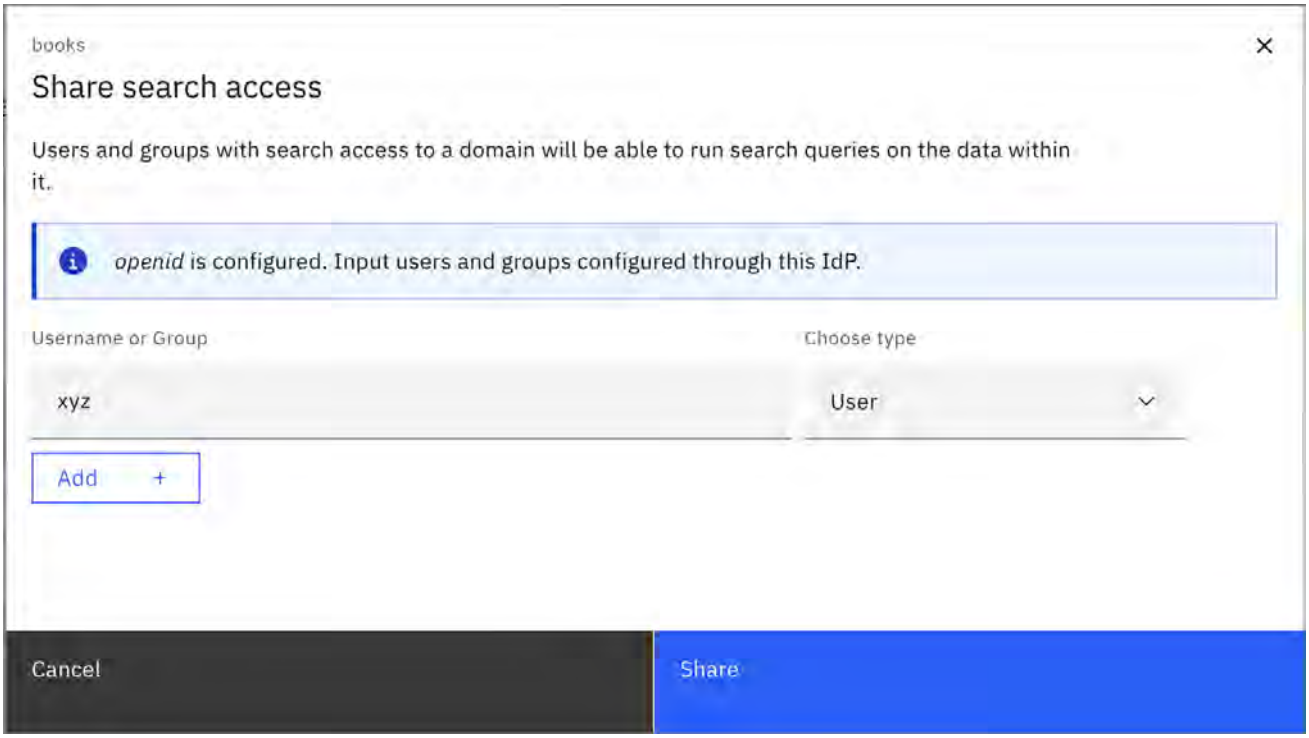
Configuring CAS Resource Access Control (CRAC)

Configure users and groups that can access specific Content-Aware Storage (CAS) domains by using the CRAC custom resource.

You can define users and groups that can have access to a domain by using CRAC API Custom Resource (CR).

To configure CAS to restrict specific users to specific CAS domains, follow these steps:

1. Open the IBM Fusion UI.
2. Go to Content-aware Storage > Domains.
3. Click the domain name that you want to open.
4. Click Actions and select Share search access.
   The Share search access dialog opens.



5. In the Username or Group field, add the username or group name to which you want to grant the domain access.
6. From the Choose type dropdown, select the type as User or Group.
   Tip: To add all users or groups at the same time, click Add.
7. Click Share.
   Note: When IDP CR is configured on your system, you can provide the external IDP token to the query search API for semantic search execution.

Granting global access to all authenticated users

To simplify access control, you can grant domain access to all authenticated users at once by using the `cas-all-authenticated` special group. This approach removes the need to add individual users or groups.
Important: The `cas-all-authenticated` group grants access to any user with a valid OpenShift® or identity provider (IDP) authentication token. Specific user and group assignments provide more granular security control and are recommended when restricted access is required.

Removing search access

You can remove users and groups that have access to specific domains.

To remove users or groups from specific CAS domains, follow these steps:

1. Open the IBM Fusion UI.
2. Go to Content-aware Storage > Domains.
3. Click the domain name that you want to open.
4. Click Actions and select Remove search access.
   The Remove search access dialog opens.

5. Select the users or groups whose search access is to be removed.
6. Click Remove.
   Removed users and groups lose authorization to access the query search API for the domain.

Configuring CAS to connect to your Identity Provider (IDP)

An Identity Provider is a system or service that creates, maintains, and manages identity information for users and provides authentication services to applications or other systems. It acts as a trusted third party that authenticates users and shares their identity information with other services.

About this task

An IDP can be integrated with CAS to enhance security, access control, and compliance. It helps CAS by providing:

- **Secure Access**: It ensures that only authorized users access sensitive content.
- **Single Sign-On (SSO)**: It enables you to log in once to access multiple systems.
- **Compliance**: It helps meet regulatory requirements with audit trails and access controls.
- **User Management**: It simplifies onboarding, offboarding, and role-based access.

Configure CAS with client ID and client secret

- Client ID: It is the ID created in the IDP to allow CAS API access.
- Client secret: It is the password that is associated with the client ID for secure access to IDP APIs.

To enable user and group validation through the Security Identity Manager (SIM) client, CAS uses the client ID and client secret to authenticate with the IDP.
To obtain the client ID and client secret, perform the following steps:

1. Create and enable API access in your IDP.
2. Configure CAS with the credentials that are provided by the IDP, as the client ID and client secret.
   CAS then uses this client ID and client secret to query the user and group APIs from the IDP.

To connect your IDP to the CAS deployment, perform the following steps:

Procedure

1. Configure an IDP secret for CAS by creating a secret that is named `idp-sec` in CAS installation namespace with `client_id` and `client_secret` in Base64 format:

```
kind: Secret
apiVersion: v1
metadata:
  name: idp-sec
  namespace: ibm-cas
data:
  client_id: <client ID in base64 format>
  client_secret: <client secret in base64 format>
type: Opaque
```

2. Create a CAS `IdentityProvider` CR to specify the custom configuration details of your IDP. Specify the following fields based on your IDP configurations:
   - email: It is the list of claims whose value must be used as the email address.
   - groups: It is the list of claims whose value must be used as groups.
   - name: It is the list of claims whose value must be used as name.
   - sub: It is the list of claims whose value must be used as subject.
   - issuer: It is the issuer URL of the external IDP. For example: https://keycloak-keycloak-automation.apps.com/auth/realms/master
   - groupUrl: It is the IDP URL for retrieving all groups. For example: https://keycloak-keycloak-automation.apps.com/auth/admin/realms/master/groups

- userUrl: It is the IDP URL for retrieving all users. For example: https://keycloak-keycloak-automation.apps.com/auth/admin/realms/master/users

```
apiVersion: cas.isf.ibm.com/v1beta1
kind: IdentityProvider
metadata:
  name: identityprovider-sample
  namespace: ibm-cas
spec:
  identityProvider:
    claims:
      email:
        - <email>
      groups:
        - <groups>
      name:
        - <preferred_username>
      sub:
        - <subject>
    issuer: <Issuer URL of IDP>
    mappingMethod: claim
    name: openid
    scimClient:
      clientSecretName: <OpenShift secret storing client_id and client_secret parameters of IDP client>
      groupUrl: <IDP URL to retrieve all groups>
      userUrl: <IDP URL to retrieve all users>
    type: OpenidProvider
```

3. To enable file-level security, add the following claims of the external IDP in the `claims` section:

```
spec:
  identityProvider:
    claims:
      groupID:
        - gidNumber
      supplementalGroups:
        - supplementalGroups
      userID:
        - uidNumber
```

Configuring file-level access control in CAS

File-level security in Content-Aware Storage (CAS) ensures that users can access only the data they are authorized to read from the file system.

Overview

File-level security uses an external identity provider (IDP) integrated with a directory server to retrieve your numeric user ID, numeric group ID, and supplemental group IDs. At the time of query, it performs a real-time access check to verify whether the user has the read permission for the parent file of each matching vector.

When file-level security is enabled, users and groups that are authorized through CAS Resource Access Control (CRAC) can still run CAS queries. However, query results are limited to files for which the user has 'read' access. File-level security is configured at the CAS deployment level and applies to all document processors and domains, both existing and newly created.

This feature requires CAS to be integrated with an external IDP that federates users from the same LDAP directory that is used to authenticate file system users.
Note: Without file-level security configuration, CAS returns results from all ingested data within a domain to any user who has domain-level access, regardless of file-level read permissions.

Before you begin

File-level access control in CAS requires the following setup:

- A directory server, such as OpenLDAP must store information about a user's numeric user ID and numeric group ID, and supplemental group IDs.
- An external Identity Provider (IDP) that uses OpenID standard, such as Keycloak. The IDP must use the directory server for user federation to retrieve and provide the following user attributes to the `/userinfo` endpoint:
  - `userID`
  - `groupID`
  - `supplementalGroups`
    Note: The `supplementalGroups` must contain the numeric IDs for the user's groups, such as `userID` and `groupID`.
    Note: The attribute names (`userID`, `groupID`, and `supplementalGroups`) are used as examples. You can use custom attribute names in the directory server. During the claim-mapping configuration (in the CRAC CR), these custom attribute names are mapped to the standard names recognized by CAS: `userID`, `groupID`, and `supplementalGroups`.
- File system clients
  - File system clients remote mount the storage file system of the CAS data source. For more information, see Mounting a remote GPFS file system in IBM Storage Scale documentation.
  - The file system client machines are then configured to use the same directory server for user authentication. For more information, see Setting up authentication servers to configure protocol user access in IBM Storage Scale documentation.
- CAS
  - Configure CAS to connect to the IDP that is configured to federate users from the same directory server used by the file system client machines. For more information, see Configuring CAS to connect to your Identity Provider (IDP).

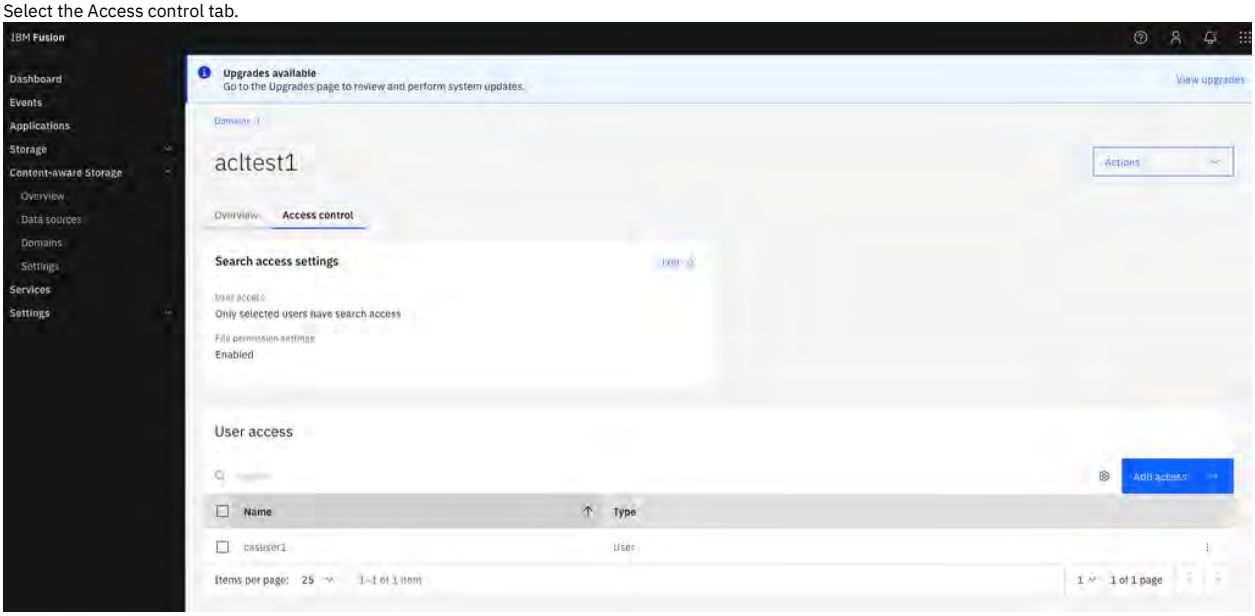
Modifying file-level access control for a domain

Using command line

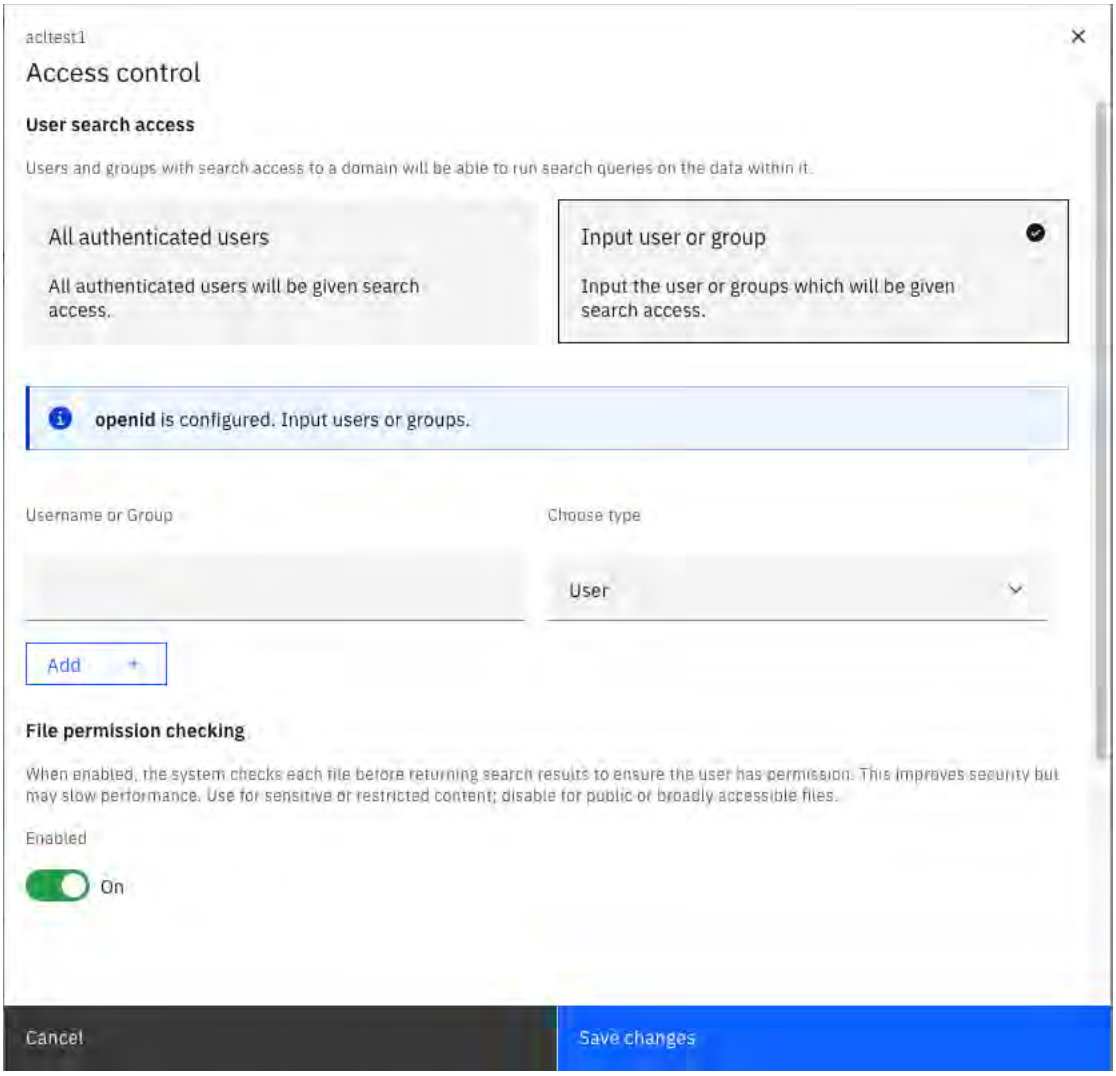
To enable or disable file-level permissions for a specific domain, add **`fileACL: true/false`** under the **`spec`** section in the **`CasResourceAccessControl`** instance for the domain. For more information, see Create a CAS Resource Access Control (CRAC) CR.

Using user interface

1. Open the IBM Fusion UI and go to Content-aware Storage > Domains.
2. Select the Domain to modify.
3. Select the Access control tab.



4. In the Search access settings section, select Edit.
5. Use the toggle to enable or disable File permission checking.



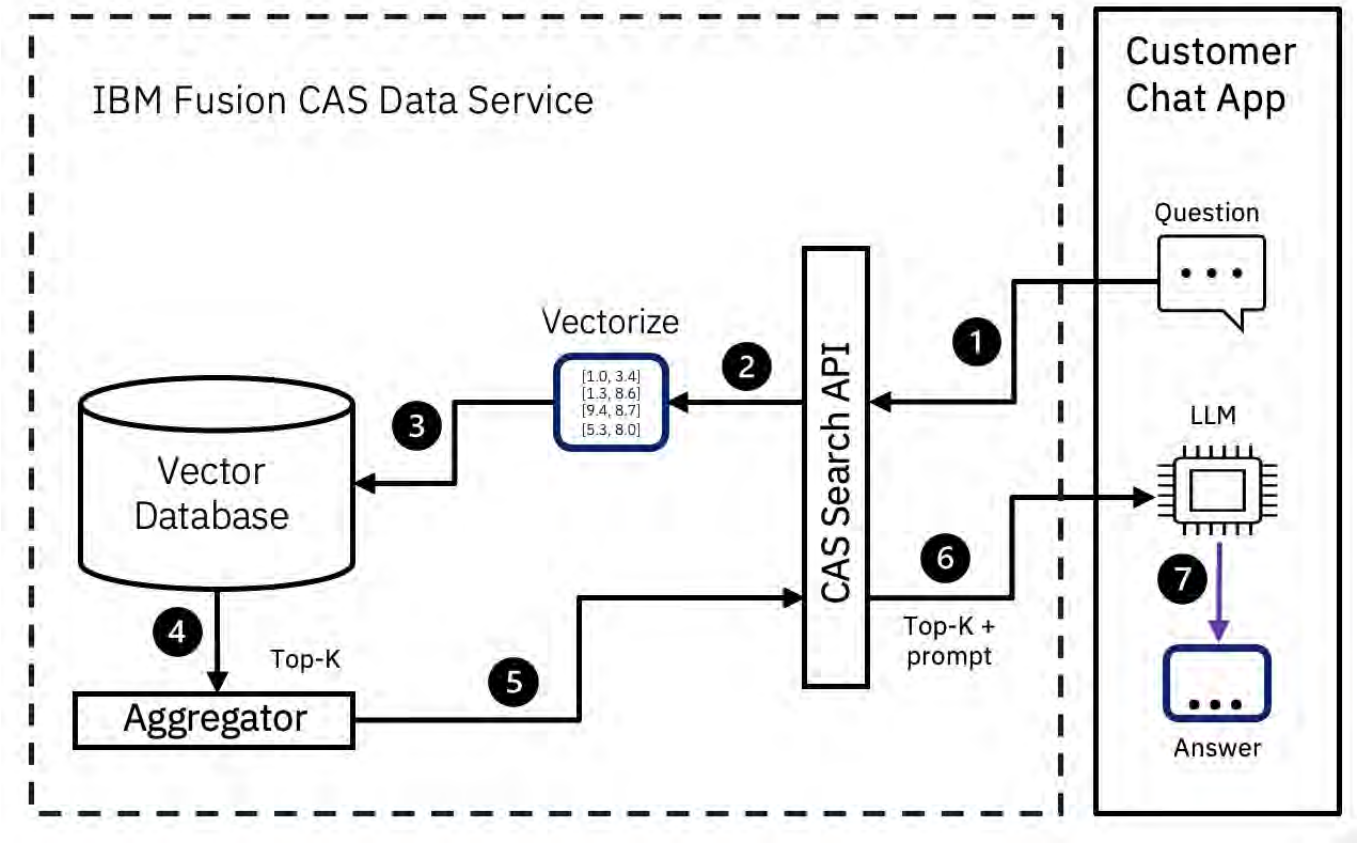
Client application with Content-Aware Storage (CAS)

You are required to provide a client application that uses the CAS search API to use the data ingested by CAS as part of your RAG application. This topic outlines the workflow for querying the CAS database using the exposed API.

Before you begin, ensure the following prerequisites are met for the client application:

- NVIDIA llama-3.1 NIM and GPU support.
- Network connectivity to issue REST APIs to CAS.

The following diagram illustrates the workflow where a user client application runs query REST calls on the CAS database and displays the results in the front end.



The entire workflow is divided into two parts: One part retrieves results from CAS, while the other obtains results from a Large Language Model (LLM).

The Vector database within the CAS namespace is designed to store data ingested from various files. The namespace also includes the query service pod, which features a FAST API (REST API). To retrieve information from the Vector database, there exists an installation of a document processing engine outside the namespace. Document processing is available with IBM Docling Multimodal services or NVIDIA NeMo Retriever Extraction microservices. NVIDIA re-ranking service is an optional search component that can be added to provide more accuracy. Your task involves developing an application (client application), that communicates with this REST API by using the OpenShift® cluster authentication. For more information about the FAST API methods, syntax, description, or to try the APIs, see Content-Aware Storage (CAS) APIs.

In addition, you must have a large language model of your choice that is accessible from your application. If you want to use the IBM Watson granite model LLM, then to download and use the different IBM Granite models, see IBM Granite.

When the user of the client application raises a question from its front end, a query REST Call is sent to the query service pod through the FAST API REST call. The query service in turn goes with the query to the document processing engine to get the embedding. This service generates embeddings for the data, enabling the query service to understand and process the request of the user effectively. After the results are fetched, a SQL query is sent with the embedding to the database for top "K" number of Vector results. When the user runs query from the front end, they can specify the value of this "K". For example, `Select, column from table order by, limit by vector, with embedding`. The top "K" results are returned to the client application which in turn sends it to the LLM. While embeddings make the retrieval process efficient, LLMs add a layer of contextual understanding, transforming raw data into meaningful information. Finally, the end results are published on the front end of the client application.

- **Content-Aware Storage (CAS) APIs**
  List of all REST API references for CAS service.
- **Integrating Model Context Protocol (MCP) with CAS**
  Content-Aware Storage (CAS) supports Model Context Protocol (MCP), enabling AI chat applications to access and search vector stores through natural language interactions.
- **Integrating CAS with watsonx Orchestrate by using MCP**
  Large language models move from experimentation into production. Enterprises build AI agents that require access to real-time business data. watsonx Orchestrate provides conversational AI for workflow automation. Integrating watsonx Orchestrate with enterprise systems such as Content-Aware Storage (CAS) often requires custom integration code for each data source.

Content-Aware Storage (CAS) APIs

List of all REST API references for CAS service.

/cas/api/v1

For more information about Swagger docs, see **`https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/docs#`**.

Authentication
: For authentication of REST APIs, ensure that you have a valid OpenShift® Container Platform bearer token in the Authorization header:
Retrieve the token:

1. In the OpenShift console, click your profile name and select Copy login command.

2. Log in with your credentials. Then click the Display Token link.
3. Copy the token value available under the Log in with this token section.
   Note: For authentication of REST APIs when an external Identity Provider (IDP) is configured, ensure that you have a valid IDP bearer token in the Authorization header.

Swagger documentation

1. In your browser, go to Swagger: **`https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/docs#`**.
   The Swagger document specifies the list of resources that are available in the REST API and the operations that can be called on those resources. The Swagger document also specifies the list of parameters for an operation, including the name and type of the parameters and information about acceptable values for those parameters.

2. In the FastAPI page, click Authorize.
3. In the Available authorizations window, enter a value for HTTPBearer.
4. Click Authorize.

Try APIs from Swagger

In your browser, go to Swagger: **`https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/docs#`**.
In Swagger, expand the API and click Try it out.

- **`/vector_stores`**
  - Description: It returns a document processor "vector stores" query that you have access to.
  - Method: GET
  - Query parameters:

Parameter:Type	Description
"after":string	Optional. A cursor for use in pagination. This parameter is not yet implemented.
"before":string	Optional. A cursor for use in pagination. This parameter is not yet implemented.
"limit":int	Optional. A limit on the number of objects to be returned. The default limit is 10.
"order":string	Optional. Sorts order by the vector store ID - **`asc`** for ascending order and **`desc`** for descending order. Default value is **`desc`**.

  - Usage: Check Swagger.
  - Response: Check Swagger.

- **`/vector_stores/{vector_store_id}/search`**
  - Description: Matches the exact words and uses Natural Language Processing (NLP) and machine learning to find results that are semantically relevant. If you do not have access to a specific domain, the API raises a 403 (Invalid access) error. If file-level security is enabled, semantic search results are restricted to only those files, which the CAS query user is authorized to read in the IBM Storage Scale file system.
  - Method: POST
  - Path parameter:

Parameter:Type	Description
"vector_store_id":string	Required. Name of a particular document processor that ingested documents from a particular domain.

  - Request body parameters:

Parameter:Type	Description
"query":string	Required. The natural language question that is used by the embedding model to return a list of relevant snippets of text from the ingested documents.
"filters":dict	Optional. A filter to apply based on file attributes. For more information, see Filter parameters.
"max_num_results":int	Optional. The maximum number of results returned. The default value is "10".
"ranking_options":dict	Optional. Ranking options for search default is null: "ranker":str - Optional. The default value is "auto". "score_threshold":float - Optional. The default value is "0". This parameter is not yet implemented.
"rewrite_query":bool	Optional. Whether to rewrite the natural language query for vector search. This parameter is not yet implemented.
"enable_source":boolean	Optional. If true, any source information about the document is returned. Default value is "False".
"enable_content_metadata" :boolean	Optional. If true, any metadata about the relevant document is returned. Default value is "False".

  - Usage: Check Swagger.
  - Response: Check Swagger.

- **`/vector_stores/{vector_store_id}/files/{file_id}/content`**
  - Description: It returns all the content (text chunks) for a specific **`vector-store`** and file by their IDs.
  - Method: GET
  - Query parameters:

Parameter	Description
enable_source	Optional. When enabled, it returns top-level source file **`metadata`** attributes, or **`null`**, if not enabled.
enable_content_metadata	Optional. When enabled, it returns content **`metadata`** attributes, or **`null`**, if not enabled.

  - Usage: Check Swagger.
  - Response: Check Swagger.

Filter parameters

Two types of filters can be used:

- Comparison filter: A filter used to compare a specified attribute key to a defined value using a defined comparison operation.

Parameter	Type	Description
key	**`str`**	Required. The key to compare against the value.

Parameter	Type	Description
type	`str`	Required. Comparison operators: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`, `in`, `nin` The set of valid operators for a comparison filter depends on the key being compared.
value	`str`, `int`, `float`, `bool`, `ISO8601 str`	Required. The value to compare against the key, which can be an array for `in` or `nin` operators.

- Compound filter: A filter used to combine an array of filters using `and` or `or`.

Parameter	Type	Description
type	`str`	Required. A type of logical operation: `and`, `or`.
filters	`array`	Required. An array of filters to combine. Items can be Comparison filter or Compound filter.

Supported keys:

Key	Type	Description
`file_id`	`str`	The file identifier.
`file_size`	`int`, `float`	The size of the file in bytes.
`file_path`	`str`	The full path of the file including the file name.
`file_name`	`str`	The full name of the file after the last path delimiter.
`file_type`	`str`	The type of source file as determined by the ingester (for example: PDF).
`file_created`	`ISO8601 str`	The creation date of the file in ISO8601 format.
`file_modified`	`ISO8601 str`	The last modified date of the file in ISO8601 format.
`attributes`	`str`, `int`, `bool`	The custom attributes set on the file through IBM Data Cataloging service tags. For more information, see Managing tags.

Supported comparison operators for key types:

Key type	Operators
`str`	`eq`, `ne`, `in`, `nin`, `contains`
`int`, `float`	`eq`, `ne`, `gt`, `gte`, `lt`, `lte`, `in`, `nin`
`ISO8601 str`	`eq`, `ne`, `gt`, `gte`, `lt`, `lte`, `in`, `nin`
`bool`	`eq`, `ne`

Note: When using the `in` or `nin` operators, the `value` parameter of the comparison must be an array of the appropriate type.
Examples

- Comparison filter:

```
"filters": {"key":"file_name","type":"eq","value":"sg248497.pdf"}
```

- Compound filter:

```
"filters": {
    "type": "and",
    "filters": [ {"key":"file_path","type":"eq","value":"/gpfs/fs1/gt20"},
                 {"key":"file_size","type":"lt","value":28000000},
                 {"key":"file_size","type":"gt","value":24000000}
               ]
}
```

- Compound filters support two nesting levels:

```
"filters": {
    "type":"or",
    "filters": [ {"key":"file_size","type":"gt","value":2000000},
                 {"type":"and",
                  "filters": [ {"key":"date_created","type":"gt","value":"2021-05-01"},
                               {"key":"date_created","type":"lt","value":"2021-05-31"}
                             ]
                 }
               ]
}
```

/api/v1/querysearch

Note: The `/api/v1/querysearch` APIs are deprecated with IBM Fusion 2.12.0 (CAS 1.1.0) release.

Authentication

For authentication of REST APIs, ensure that you have a valid OpenShift Container Platform bearer token in the Authorization header:
Retrieve the token:

1. In the OpenShift console, click your profile name and select Copy login command.
2. Log in with your credentials. Then click the Display Token link.
3. Copy the token value available under the Log in with this token section.
   Note: For authentication of REST APIs when an external Identity Provider (IDP) is configured, ensure that you have a valid IDP bearer token in the Authorization header.

Swagger documentation

1. In your browser, go to Swagger: `https://<fusionconsoleurl>/api/v1/querysearch/docs`.
   The Swagger document specifies the list of resources that are available in the REST API and the operations that can be called on those resources. The Swagger document also specifies the list of parameters for an operation, including the name and type of the parameters and information about acceptable values for those parameters.
2. In the FastAPI page, click Authorize.
3. In the Available authorizations window, enter a value for HTTPBearer.
4. Click Authorize.

Try APIs from Swagger

In your browser, go to Swagger: `https://<fusionconsoleurl>/api/v1/querysearch/docs`.

In Swagger, expand the API and click Try it out.

- `/tables`
  - Description: It returns a document processor "tables" query that you have access to.
  - Method: GET
  - Usage: Check Swagger.
  - Response: Check Swagger.
- `/semantic_search`
  - Description: Matches the exact words and uses Natural Language Processing (NLP) and machine learning to find results that are semantically relevant. If you do not have access of a specific domain, the API raises a 403 (Invalid access) error. If file-level security is enabled, semantic search results are restricted to only those files, which the CAS query user is authorized to read in the IBM Storage Scale file system.
  - Method: POST
  - Parameters:

Parameter: Type	Description
"query": string	The natural language question that is used by the embedding model to return a list of relevant snippets of text from the ingested documents.
"table": string	Name of a particular document processor that ingested documents from a specific domain.
"limit": int	The maximum number of results returned, default is 5.
"enable_source": boolean	If true, any source information about the document is returned.
"enable_content_metadata": boolean	If true, any metadata about the relevant document is returned.

  - Usage: Check Swagger.
  - Response: Check Swagger.

End-user CLI chatbot sample

The CLI-powered chatbot enables a secured, authenticated access to Content-Aware Storage (CAS) through a conversational CLI. It is powered by LLMs such as IBM watsonx Granite, Open AI, NVIDIA NIM, and Ollama. It uses CAS's semantic APIs to deliver efficient, AI-enabled insights, accelerating RAG workflows with improved performance, security, and efficiency.

For more information, follow the reference example in GitHub.

Access Control List (ACL) users control the query search process. The system retrieves user identities from the Identity Provider (such as Keycloak) and validates them against CAS ACL users based on roles and permissions. If validation succeeds, the system issues a token, which the user uses to run the CAS query API.

```
                    CAS ENTERPRISE CHATBOT
                    v1.0.0 - Enhanced Edition

╭──────────────────────────────── ABOUT CAS ────────────────────────────────╮
│ Content-Aware Storage powers AI applications like RAG with faster insights, lower costs, better performance, stronger security, and simplified operations. │
╰──────────────────────────────────────────────────────────────────────────╯


[17:51:23] INFO     Configuration loaded from config.yaml
           INFO     CAS Chatbot initialized

Starting CAS Chatbot...
🔐 Authenticating with OpenShift...
Authentication successful
[17:51:29] INFO     Successfully authenticated with OpenShift
🔍 Checking CAS service status...
Query Search API: OK
   Welcome to IBM Fusion Content Aware Storage
Health Check: OK
[17:51:30] INFO     CAS service status check passed
```

```
[fusion-assist-sre] 👤 You: help
                    Available Commands
┌──────────────────┬──────────────────────────┬──────────────────────────┐
│ Command          │ Description              │ Example                  │
├──────────────────┼──────────────────────────┼──────────────────────────┤
│ help             │ Show this help menu      │ help                     │
│ bye/exit/quit    │ Exit the chatbot         │ bye                      │
│ history          │ Show query history       │ history                  │
│ clear            │ Clear query history      │ clear                    │
│ tables           │ List available tables    │ tables                   │
│ table <name>     │ Switch to a specific table │ table customers        │
│ stats            │ Show session statistics  │ stats                    │
│ export           │ Export chat history      │ export                   │
│ <query>          │ Ask a question           │ What are the sales trends? │
└──────────────────┴──────────────────────────┴──────────────────────────┘
[fusion-assist-sre] 👤 You: table gt20
Switched to table: gt20
[gt20] 👤 You: list down research papers.
⋮ 🔍 Searching...
[17:55:07] INFO     Query executed successfully for table: gt20
Trying LLM provider: nvidia
Based on the provided data, I found the following research papers relevant to the topic:

1. IBM Redbook: IBM System Storage SAN Volume Controller and Storwize V7000 Best Practices and Performance Guidelines, SG24-7521
2. IBM Redbook: Implementing IBM Spectrum Virtualize for Public Cloud Version 8.3, REDP-5466
3. IBM Redbook: Implementing the IBM Storwize V7000 with IBM Spectrum Virtualize V8.2.1, SG24-7938
4. IBM Redbook: Implementing the IBM System Storage SAN Volume Controller with IBM Spectrum Virtualize V8.1, SG24-7933
5. IBM Redpaper: IBM Storage for Red Hat OpenShift Container Platform V3.11 Blueprint Version 1 Release 1, REDP-5564
6. IBM Redpaper: IBM Storage for Red Hat OpenShift Blueprint Version 1 Release 4, REDP-5565
7. IBM Redpaper: Red Hat OpenShift and IBM Cloud Paks on IBM Power Systems: Volume 1, SG24-8459
8. IBM Knowledge Center: IBM Real-time Compression and performance guidelines summary, REDP-5466
9. IBM Knowledge Center: IBM Disk Magic, REDP-5466
10. IBM Knowledge Center: IBM Storage for Red Hat OpenShift Container Platform V3.11 Blueprint Version 1 Release 1, REDP-5564

Please note that these papers are based on the provided data and might not be the most relevant or up-to-date research papers.
[gt20] 👤 You:
```

Integrating Model Context Protocol (MCP) with CAS

Content-Aware Storage (CAS) supports Model Context Protocol (MCP), enabling AI chat applications to access and search vector stores through natural language interactions.

Model Context Protocol (MCP)

Model Context Protocol (MCP) is a standardized JSON-RPC 2.0 protocol that allows AI assistants and chat applications to interact with external services. MCP enables AI clients to discover available capabilities and use them through conversational interfaces.

Benefits of MCP server

- AI assistants can discover and use CAS vector stores automatically.
- It supports natural language queries for vector search operations.
- It provides both standard and streaming response options.

Available MCP tools

CAS uses two primary tools through the MCP interface for working with vector stores.

list_vector_stores

- Description: Lists all vector stores available in the CAS instance. This tool retrieves the complete inventory of vector stores that can be accessed for search operations.
- Parameters: The parameters (`server_url` and `auth_token`) are needed as input for the CAS server you want to search in.

Parameter	Type	Description
`server_url`	string	Required. Base URL of the CAS server (for example, `https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>`).
`auth_token`	string	Required. Authentication token for API access.

- Usage example: An AI assistant calls this tool to discover which vector stores are available before performing searches.

search_vector_stores

- Description: Performs semantic search within a specified vector store. This tool uses vector embeddings to find content similar to the provided query text that returns ranked results based on semantic similarity.
- Parameters:

Parameter	Type	Description
`server_url`	string	Required. Base URL of the CAS server (for example, `https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>`).
`auth_token`	string	Required. Authentication token for API access.
`vector_store_id`	string	Required. ID of the vector store to search.
`query`	string	Required. Search query text.
`limit`	integer	Optional. Maximum number of results to return. The default value is '10'.

- Usage example: To find documents about 'machine learning best practices' in a vector store named 'technical_docs', the AI assistant calls this tool with the query text and store ID. The tool returns semantically similar documents that are ranked by relevance score.

get_vector_store_file_content

- Description: Retrieves the complete content of a specific file from a vector store. Use this tool when you need the complete document after finding it through search results. The `file_id` can be obtained from `search_vector_stores` results.
- Parameters:

Parameter	Type	Description
`server_url`	string	Required. Base URL of the CAS server (for example, `https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>`).
`auth_token`	string	Required. Authentication token for API access.
`vector_store_id`	string	Required. ID of the vector store to search.
`file_id`	string	Required. The ID of the file to fetch content for.

- Usage example: After searching for `storage configuration` and finding relevant results, use this tool with the `file_id` from those results to fetch the complete document content for detailed analysis.
- Example: Standard MCP endpoint

```
curl -k -N -X POST "https://<console-ibm-spectrum-fusion-ns.apps.openshift>/cas/api/v1/mcp" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -d '{
    "jsonrpc": "2.0",
    "id": 6,
    "method": "tools/call",
    "params": {
      "name": "get_vector_store_file_content",
      "arguments": {
        "server_url": "https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>",
        "auth_token": "your_auth_token",
        "vector_store_id": "your_store_id",
        "file_id": "5243905"
      }
    }
  }'
```

- Example: MCP streamable endpoint

```
curl -k -N -X POST \ https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/mcp-streamable/ \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{
    "jsonrpc": "2.0",
    "id": 6,
    "method": "tools/call",
    "params": {
      "name": "get_vector_store_file_content",
      "arguments": {
        "server_url": "https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>",
        "auth_token": "your_auth_token",
        "vector_store_id": "your_store_id",
        "file_id": "5243905"
      }
    }
  }'
```

Authentication

For more information on authentication, see Content-Aware Storage (CAS) APIs.

All MCP tool calls require authentication. Provide the CAS authentication token in the `auth_token` parameter when you call any tool. The token is validated by using the same authentication system as the REST APIs.

MCP endpoints

/cas/api/v1/mcp

- Description: Standard MCP endpoint that provides JSON-RPC 2.0 protocol support for synchronous operations.
- Method: POST
- Content-Type: `application/json`

Initialize example:

```
curl -X POST \
  https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/mcp \
  -H "Content-Type: application/json" \
  -d '{
    "jsonrpc": "2.0",
    "id": 1,
    "method": "initialize",
    "params": {}
  }'
```

List available tools:

```
curl -X POST \
  https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/mcp \
  -H "Content-Type: application/json" \
  -d '{
    "jsonrpc": "2.0",
    "id": 2,
    "method": "tools/list",
    "params": {}
  }'
```

Call MCP tools:

```
curl -X POST \
  https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/mcp \
  -H "Content-Type: application/json" \
  -d '{
    "jsonrpc": "2.0",
    "id": 3,
    "method": "tools/call",
    "params": {
      "name": "list_vector_stores",
      "arguments": {
        "server_url": "https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>",
        "auth_token": "your_auth_token"
      }
    }
  }'
```

/cas/api/v1/mcp-streamable/

- Description: Streaming MCP endpoint that provides real-time response delivery by using Server-Sent Events (SSE). Use this endpoint when continuous or progressive responses are needed.
- Method: POST
- Content-Type: `application/json`
- Accept: `text/event-stream`

Initialize example:

```
curl -N -X POST \
  https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/mcp-streamable/ \
  -H "Content-Type: application/json" \
  -H "Accept: text/event-stream" \
  -d '{
    "jsonrpc": "2.0",
    "id": 1,
    "method": "initialize",
    "params": {
      "protocolVersion": "2024-11-05",
      "capabilities": {},
      "clientInfo": {
        "name": "test-client",
        "version": "1.0.0"
      }
    }
  }'
```

Call streaming tools:

```
curl -k -N -X POST \
https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/mcp-streamable/ \
-H "Content-Type: application/json" \
-H "Accept: application/json, text/event-stream" \
-d '{
  "jsonrpc": "2.0",
  "id": 3,
  "method": "tools/call",
  "params": {
    "name": "list_vector_stores",
    "arguments": {
      "server_url": "https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>",
      "auth_token": "your-token-here"
    }
  }
}'
```

```
curl -N -X POST \
  https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>/cas/api/v1/mcp-streamable/ \
  -H "Content-Type: application/json" \
  -H "Accept: text/event-stream" \
  -d '{
    "jsonrpc": "2.0",
    "id": 4,
    "method": "tools/call",
```

```
    "params": {
      "name": "search_vector_stores",
      "arguments": {
        "server_url": "https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>",
        "auth_token": "your_auth_token",
        "vector_store_id": "your_store_id",
        "query": "your search query",
        "limit": 10
      }
    }
  }'
```

Getting started using MCP endpoints in your application

To use MCP with CAS, follow these steps:

1. Obtain the CAS server URL and authentication token.
2. Configure the AI chat client to connect to the MCP endpoint.
   The AI assistant discovers the available tools automatically.
3. Begin with using natural language queries that require information from vector stores.

Error handling

MCP follows JSON-RPC 2.0 error conventions. If a request fails, the response includes an error object with a code and message.

Common error codes:

- -32700: Parse error (invalid JSON)
- -32600: Invalid request
- -32601: Method not found
- -32602: Invalid parameters
- -32603: Internal error

Integrating CAS with watsonx Orchestrate by using MCP

Large language models move from experimentation into production. Enterprises build AI agents that require access to real-time business data. watsonx Orchestrate provides conversational AI for workflow automation. Integrating watsonx Orchestrate with enterprise systems such as Content-Aware Storage (CAS) often requires custom integration code for each data source.

Model Context Protocol (MCP) removes this requirement. MCP provides a standardized, conversational interface that allows AI agents to discover and invoke enterprise capabilities without custom integration code. For more information about integrating MCP with CAS, see Integrating Model Context Protocol (MCP) with CAS.

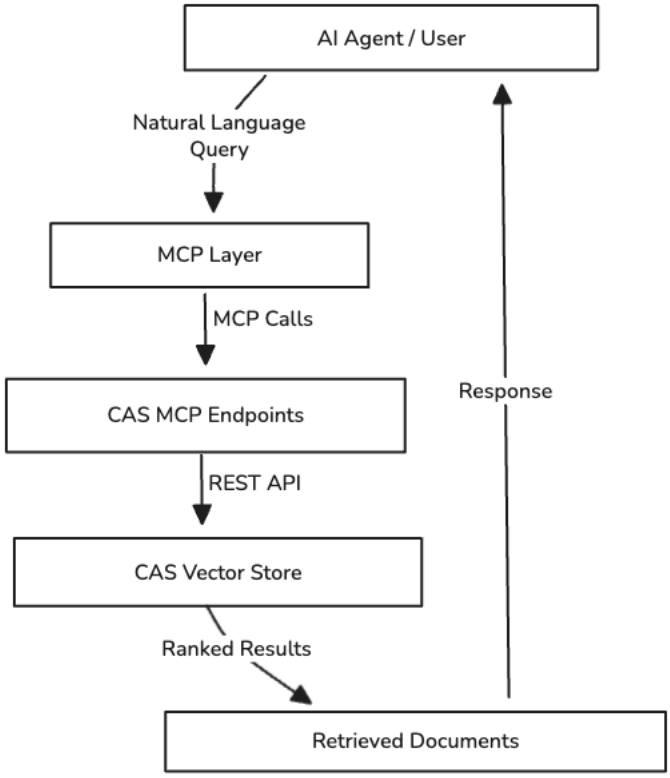
The following section explains how MCP enables watsonx Orchestrate to access CAS, describes the technical configuration, and demonstrates the integration through natural language queries.

Integration challenges

Traditional integrations between AI agents and enterprise systems introduce several challenges:

- Each vector database integration requires a custom API client code.
- AI agents cannot discover new capabilities without redeployment.
- Authentication, error handling, and connection management increase development overhead.
- Adding multiple data sources multiplies integration complexity.

Integration with watsonx Orchestrate



Validation of the architecture uses watsonx Orchestrate. The workflow follows three phases:

1. Connect CAS to watsonx Orchestrate by using MCP.
2. Create an AI agent that uses CAS tools.
3. Validate queries through natural language interaction.

Before you begin

Ensure that the following components are available:

- watsonx Orchestrate instance: An active watsonx Orchestrate deployment with CLI access. For more information about setting up this deployment, see Installing watsonx Orchestrate service.
- Model configured in watsonx Orchestrate: A model enabled to run AI agents that query CAS. For more information about adding a model, see Enabling a specific model.
- IBM Fusion instance with CAS deployed: CAS running on OpenShift®. For more information about installing CAS, see Installing Content-Aware Storage (CAS).
- CAS authentication token: Bearer token for CAS API access.
- CAS URL: The IBM Fusion console URL of the cluster. For example: `https://<console-ibm-spectrum-fusion-ns.apps.openshifturl>`
- Network connectivity: Network access from watsonx Orchestrate instance to the CAS server route.

Configuration

1. Set up the watsonx Orchestrate CLI environment.
   The watsonx Orchestrate CLI manages MCP servers, toolkits, and agents.

   a. Download and install the watsonx Orchestrate CLI for your operating system. For more information, see https://developer.watson-orchestrate.ibm.com/getting_started/installing.
   b. Verify the installation:

   ```
   orchestrate --version
   ```

   c. Set up a Python virtual environment (recommended):

   ```
   python3 -m venv orchestrate-env
   source orchestrate-env/bin/activate  # On Windows: orchestrate-env\Scripts\activate
   ```

2. Configure the watsonx Orchestrate environment.
   a. In the watsonx Orchestrate web console, retrieve the watsonx Orchestrate instance URL from Profile Settings. Add and activate the environment:

   ```
   orchestrate env add -n wxo-instance -u <your orchestrate url>

   orchestrate env activate wxo-instance
   ```

   b. When prompted, enter the watsonx Orchestrate credentials. Credentials can be obtained from the watsonx Orchestrate documentation.
3. Configure the CAS MCP server URL.
   a. To construct the URL, append `/cas/api/v1/mcp-streamable/` to the base CAS URL described in the Before you begin section. For example:
   `https://console-ibm-spectrum-fusion-ns.apps.<openshift-cluster-domain>/cas/api/v1/mcp-streamable/`
4. Register the CAS MCP server as a toolkit in watsonx Orchestrate:

```
orchestrate toolkits add \
  --kind mcp \
  --name cas \
  --description "MCP server for CAS vector stores" \
  --url <your-cas-mcp-url> \
  --transport streamable_http \
  --tools "search_vector_stores,list_vector_stores,get_vector_store_file_content"
```

This command registers the CAS MCP server and exposes the following tools to watsonx Orchestrate agents:

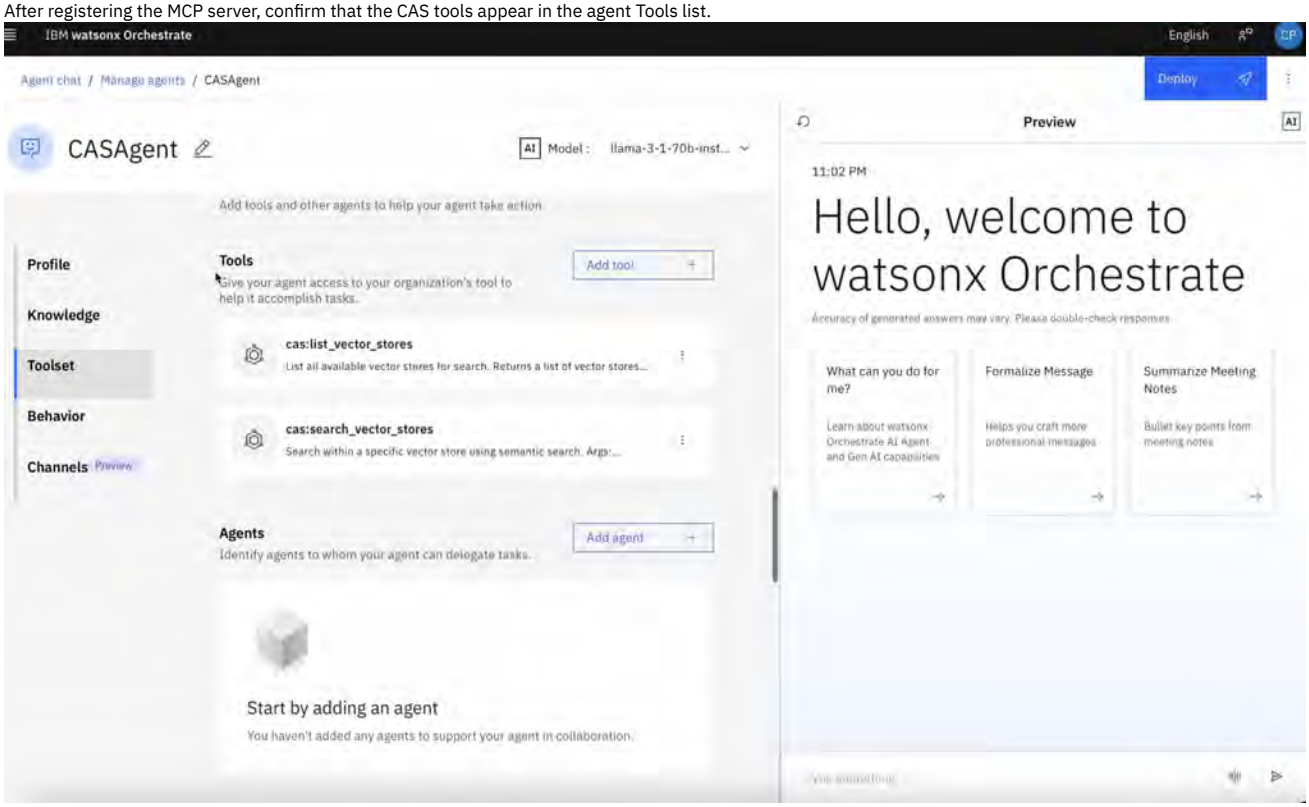
- `search_vector_stores`
- `list_vector_stores`
- `get_vector_store_file_content`

5. Create and configure the AI agent.
   a. Open the watsonx Orchestrate web console and click Create new agent.
   b. Enter the required details and click Create.
   c. In the Tools section of agent configuration, select Add Tool.
   d. Select tools from the cas toolkit.
   e. Click Add tool and add the following tools:
      - `list_vector_stores`
      - `search_vector_stores`
      - `get_vector_store_file_content`
   f. Click Save.
   g. Click Deploy.

Validation

Verify tool registration

After registering the MCP server, confirm that the CAS tools appear in the agent Tools list.



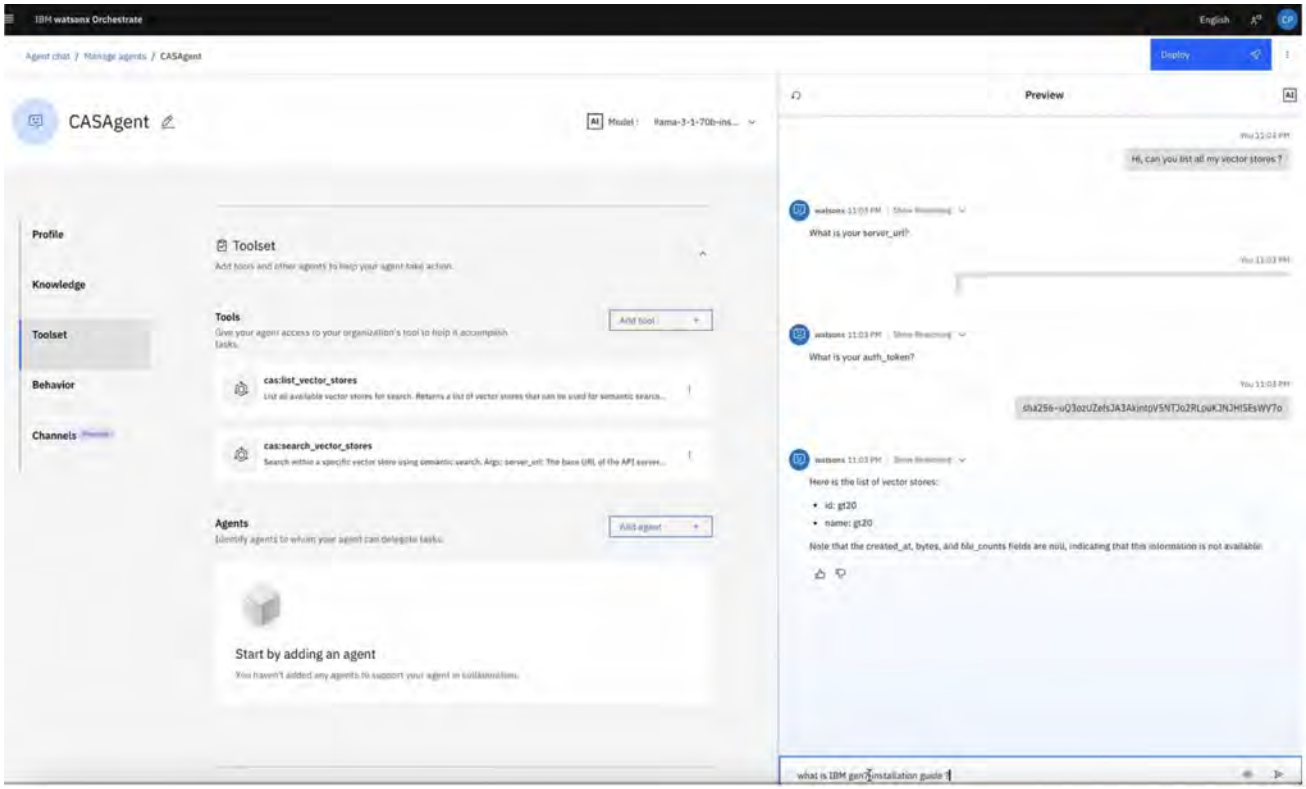
Verify tool execution

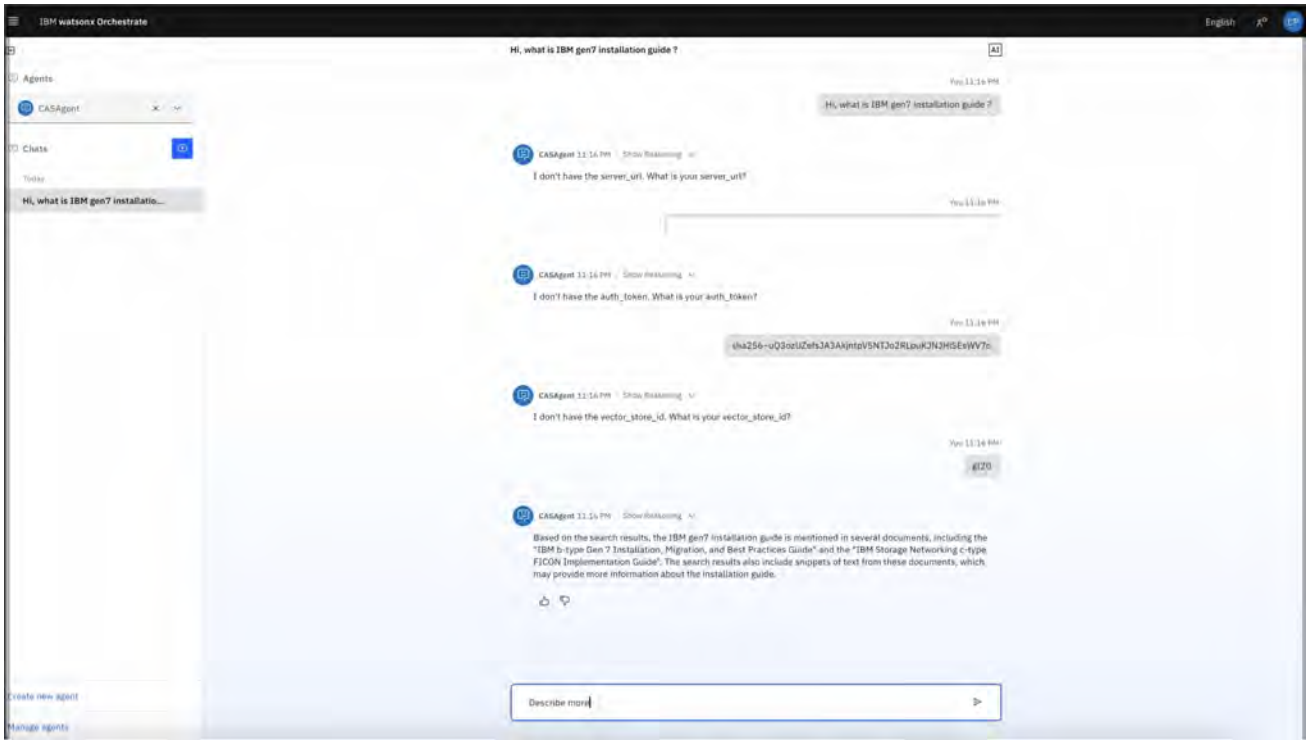
Test the deployed agent by using natural language prompts.

Example prompt:

```
What vector stores are available in CAS?
```

Note: When prompted, provide the CAS route URL and bearer token in the chat interface.





Results

The integration connects CAS with watsonx Orchestrate through MCP. AI agents access enterprise vector stores directly by using natural language.

With this configuration, you can:

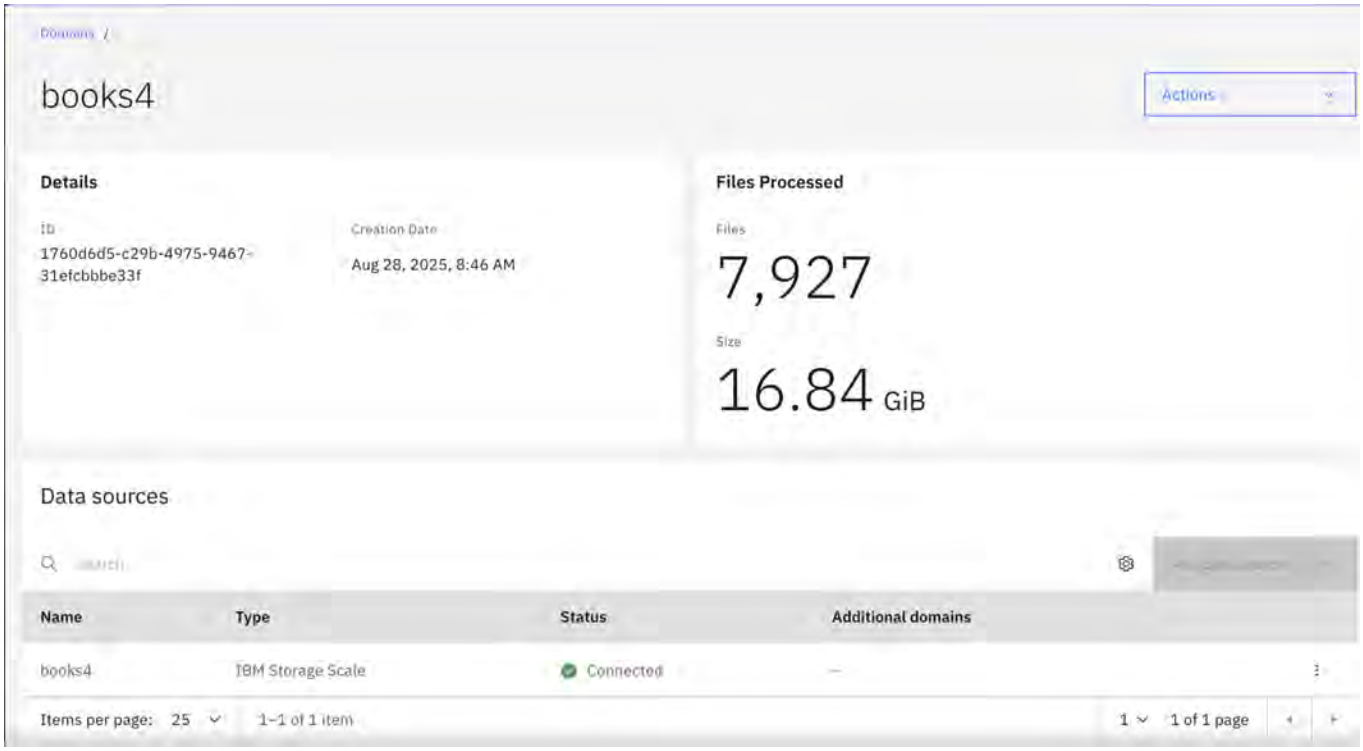
- Run agentic workflows that retrieve and act on enterprise knowledge.
- Integrate retrieval-augmented generation (RAG) pipelines with CAS vector stores.
- Eliminate custom integration code and reduce operational complexity.

Monitoring

This page provides a comprehensive view of monitoring capabilities, structured into two key sections: monitoring using dashboard in Content-Aware Storage (CAS) user interface and metric collection through Prometheus.

Monitor from CAS dashboard

Go to Content-Aware Storage > Overview page of the IBM Fusion user interface to view the CAS dashboard. It provides an inventory of domains and data sources and newly created domain entries. You can use this information to get insights of system activity.



CAS metrics

CAS provides a set of Prometheus metrics to give you insights on processing time of a batch of documents with respect to the document sizes.

Metric	Description
`cas_documents_batch_duration_seconds [label: document_processor_name, event_type]`	Overall time taken by CAS to successfully process a batch of new, updated, or deleted documents.
`cas_ingest_service_batch_duration_seconds [label: document_processor_name]`	Time taken by document processor to successfully process a batch of new or updated documents. This is a subset of the overtime time measured by `cas_documents_batch_duration_seconds`.
`cas_documents_size_bytes_total [label: document_processor_name, event_type]`	Total size of all documents in a batch of new, updated, or deleted documents. This metric is published regardless of whether the batch processing is successfully or not. It gives you insights on whether there is any unexpected spike of document size.
`cas_document_size_bytes [label: document_processor_name, event_type]`	Size of each document. This metric is published regardless of whether the batch processing is successfully or not. It gives you insights on whether there is any unexpected large file.
`cas_documents_count [label: file_type, domain, documentProcessor]`	Total number of files in the domain that the specified document processor handles. This metric is published only after the document processor processes at least one file.
`cas_documents_size_bytes [label: file_type, domain, documentProcessor]`	Total size in bytes of all files present in the domain that the specified document processor handles. This metric is published only after the document processor processes at least one file.

The following table provides label definitions:

Label	Value
`document_processor_name`	`domain name`
`event_type`	`update: new or updated documents`
	`delete: deleted documents`

Uninstalling Content-Aware Storage (CAS) service

When you do not need CAS service, uninstall it from IBM Fusion.

About this task

After you run the script, the following high-level steps occur in sequence:

- Uninstalls Apache Kafka service.
- Uninstalls the CloudNativePG (CNPG) service.
- Deletes the CAS namespace.

Procedure

1. Log in to your cluster.
2. In your browser, go to this URL - https://github.com/IBM/storage-fusion/blob/master/CAS/cas-cleanup/customer_cas_cleanup.sh.
3. Download the customer_cas_cleanup.sh file and run the script.

- **Cleaning up Content-Aware Storage (CAS) data**
  You can use CRs to clean the data source.

Cleaning up Content-Aware Storage (CAS) data

You can use CRs to clean the data source.

Clean up the data source

Clean up the data source through CLI

To delete the data source, run the following command:

```
oc delete datasource <datasource-name>
```

Note: After you delete the data source, the respective PV, PVC, Kafka Topic, and Clustered Watch Folder also get deleted.

Clean up the data source through user interface

To clean up the data source through UI, follow these steps:

1. Open the IBM Fusion UI.
2. Go to Content-aware Storage > Data sources.
3. Select the data source that you want to delete.
4. In the ellipsis overflow menu, click Remove.

Cleanup steps for an S3 data source

To clean up an S3 data source, follow these steps:

1. From the data source YAML, get the associated **filesystem-name** and **fileset-name**.
2. Log in to the IBM Storage Scale cluster.
3. Unlink the fileset with the following command:

   ```
   mmunlinkfileset <filesystem-name> <fileset-name>
   ```

4. Delete the fileset:

   ```
   mmdelfileset <filesystem-name> <fileset-name>
   ```

5. Delete the bucket configuration information:

   ```
   mmafmcoskeys <bucket-name>:<fileset-name>-exportmap delete
   ```

6. Delete the data source.

Clean up the domain

To delete the Domain, run the following command:

```
oc delete Domain <domain-name>
```

Clean up the Document Processor

To delete the Document Processor, run the following command:

```
oc delete DocumentProcessor <document-processor-name>
```

Note: After you delete the Document Processor, the respective deployment, PVC, Service, and Database Table also get deleted.

Troubleshooting issues and known limitations in Content-Aware Storage (CAS)

The common troubleshooting steps and limitations in CAS service.

CAS pods see permission denied when writing to shared PVCs during upgrade

Problem statement

After upgrading CAS from 1.1.4 to 1.1.5, two related failures share a common root cause:

- **docling-serve** RapidOCR filename mismatch (**_infer** vs **_mobile**): After the upgrade, **RapidOcrModel.__init__()** expects model files with a **_mobile** filename suffix, but the stale **huggingface-model-data** PVC contains **_infer**-suffix files from the 1.1.4 download job. A **FileNotFoundError** ends uvicorn startup.

- **vllm-embedding**, **vllm-vision**, and **cast-runtime** permission denied on model-images PVC: After the **cas-ensure-model-data-job** finishes, subsequent pods run as different UIDs and GIDs and cannot write to the shared PVC.

Cause

The **ibm-cas-setuid-setgid-scc** at priority 10 wins SCC admission and sets **fsGroup: RunAsAny**, which suppresses kubelet **fsGroup** injection. This leaves shared PVC mounts owned by **root:root**, and the relevant pods cannot write to those mounts.

When to apply

Apply during an upgrade of CAS from 1.1.4 to 1.1.5.

Resolution

SCC patch and override: Create a copy of **ibm-cas-setuid-setgid-scc** that is identical in all fields except **fsGroup: MustRunAs** (with the namespace GID range) and **priority: 11**. Setting **priority: 11** causes this SCC to beat **ibm-cas-setuid-setgid-scc** during admission. The kubelet then injects the **fsGroup** GID and chowns the volume mount, allowing **vllm** to write to /models/modules and **cast-runtime** to write to /shared.

1. Create the override SCC:

```
oc apply -f - <<'EOF'
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: ibm-cas-cast-runtime-mustrunasgroup
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
allowedCapabilities: [SETUID, SETGID]
fsGroup:
  type: MustRunAs
groups: []
users: []
priority: 11
runAsUser:
  type: MustRunAsNonRoot
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
volumes: [configMap, emptyDir, persistentVolumeClaim, secret]
EOF
```

2. Create the ClusterRole:

```
oc apply -f - <<'EOF'
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: ibm-cas-cast-runtime-mustrunasgroup-use
rules:
- apiGroups: [security.openshift.io]
  resourceNames: [ibm-cas-cast-runtime-mustrunasgroup]
  resources: [securitycontextconstraints]
  verbs: [use]
EOF
```

3. Bind the ClusterRole to the relevant service accounts:

```
oc apply -f - <<'EOF'
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: ibm-cas-cast-runtime-mustrunasgroup-use
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: ibm-cas-cast-runtime-mustrunasgroup-use
subjects:
- kind: ServiceAccount
  name: ibm-isf-cas-operator-system
  namespace: ibm-cas
- kind: ServiceAccount
  name: default
  namespace: ibm-cas
EOF
```

4. Delete the affected pods so that they restart under the new SCC:

```
oc delete pod -n ibm-cas -l app.kubernetes.io/component=cast-runtime,app.kubernetes.io/part-of=deucalion-domain
oc delete pod -n ibm-cas -l app.kubernetes.io/component=vllm
```

Important: **priority: 11** is required to beat **ibm-cas-setuid-setgid-scc** (priority 10) during SCC admission. **ibm-cas-cast-runtime-mustrunasgroup** must be bound to both **ibm-isf-cas-operator-system** (**cast-runtime**, **vllm**) and **default** (**cas-ensure-model-data-job**, first writer on the **huggingface-model-data** PVC) so that the injected **fsGroup** GID is consistent across all pods sharing those volumes.

CAS operator reconcile stalls: Missing EventManager NetworkPolicy

Problem statement

The CAS operator logs repeat the following pattern approximately every 60 seconds throughout the entire install and any subsequent reconcile:

```
Event Manager service url: https://<clusterIP>:10666/api/v1/eventmanager/alerts
Error POSTing info request (...): Post "https://<clusterIP>:10666/api/v1/eventmanager/alerts":
  context deadline exceeded (Client.Timeout exceeded while awaiting headers)
```

Each timeout adds approximately 60 seconds of dead time to every reconcile cycle, visibly slowing installation and upgrade progress.

Important: This fix is non-recurrent. Apply it once before CAS installation. It persists across installations and upgrades unless **isf-serviceability-operator** reconciles the policy away.

Cause

The **eventmanager-network-policy** in **ibm-spectrum-fusion-ns** has no ingress rule permitting traffic from the **ibm-cas** namespace to port **10666**. Every outbound POST from the CAS operator is silently dropped, and the HTTP client blocks until the OS TCP timeout expires.

When to apply

Apply this fix before CAS installation begins. Every reconcile cycle stalls without it.

Re-apply if the symptom returns after an **isf-serviceability-operator** reconcile overwrites the policy.

Resolution

Save the following manifest to a file named allow-cas-operator-to-eventmanager.yaml and apply it:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-cas-operator-to-eventmanager
  namespace: ibm-spectrum-fusion-ns
  labels:
    app.kubernetes.io/managed-by: cas-operator
spec:
  podSelector:
    matchLabels:
      app: isfemdep
  policyTypes:
  - Ingress
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          kubernetes.io/metadata.name: ibm-cas
      podSelector:
        matchLabels:
          app.kubernetes.io/component: operator
          app.kubernetes.io/name: cas.isf.ibm.com
    ports:
    - port: 10666
      protocol: TCP
```

```
oc apply -f allow-cas-operator-to-eventmanager.yaml
```

Result

After applying the policy, the CAS operator logs no longer show `context deadline exceeded` for Event Manager POST calls.

Docling, vllm, and cast-runtime pods crash with permission errors: SCC fsGroup override

Problem statement

After a CAS upgrade or on a fresh install, one or more of the following symptoms occur:

- The **docling** pod shows **1/2 CrashLoopBackOff**. Logs show:

  ```
  FileNotFoundError: .../ch_PP-OCRv4_det_mobile.onnx does not exist
  ```

  A stale **huggingface-model-data** PVC contains **_infer**-suffix model files. CAS 1.1.5 expects **_mobile**-suffix files.

- The **vllm-embedding** or **vllm-vision** pod shows **1/2 CrashLoopBackOff**. Logs show:

  ```
  PermissionError: [Errno 13] Permission denied: '/models/modules'
  ```

- The **cast-runtime** pod logs show:

  ```
  Error publishing messages [Errno 13] Permission denied: '/shared/<jobID>'
  ```

  Inspecting the volume mount inside the pod confirms root ownership:

  ```
  $ stat /shared
  Access: (0755/drwxr-xr-x)  Uid: (0/root)  Gid: (0/root)
  ```

Important: This fix is non-recurrent. Apply it once before the affected pods first start. It persists until the SCC or RBAC binding is removed.

Cause

The **ibm-cas-setuid-setgid-scc** is configured with **fsGroup: RunAsAny** at priority 10. When this SCC wins pod admission, the kubelet skips **fsGroup** injection and does not chown volume mounts. PVC mounts remain owned by **root:root**, and pods running as a non-root UID cannot write to them.

When to apply

Apply all steps before the **DocumentProcessor**, **vllm**, and **docling** pods first start. The SCC must exist at pod admission time.

If the affected pods have already started under the wrong SCC, complete all steps and then bounce the pods (step 4).

Resolution

1. Create an override SCC at priority 11, which takes precedence over the existing SCC at priority 10:

```
oc apply -f - <<'EOF'
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: ibm-cas-cast-runtime-mustrunasgroup
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
allowedCapabilities: [SETUID, SETGID]
fsGroup:
  type: MustRunAs
groups: []
users: []
runAsUser:
  type: MustRunAsNonRoot
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
volumes: [configMap, emptyDir, persistentVolumeClaim, secret]
EOF
```

2. Create a ClusterRole that grants use of the new SCC:

```
oc apply -f - <<'EOF'
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: ibm-cas-cast-runtime-mustrunasgroup-use
rules:
- apiGroups: [security.openshift.io]
  resourceNames: [ibm-cas-cast-runtime-mustrunasgroup]
  resources: [securitycontextconstraints]
  verbs: [use]
EOF
```

3. Bind the `ClusterRole` to the relevant service accounts:

```
oc apply -f - <<'EOF'
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: ibm-cas-cast-runtime-mustrunasgroup-use
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: ibm-cas-cast-runtime-mustrunasgroup-use
subjects:
- kind: ServiceAccount
  name: ibm-isf-cas-operator-system
  namespace: ibm-cas
- kind: ServiceAccount
  name: default
  namespace: ibm-cas
EOF
```

4. Bounce the affected pods so that they are re-admitted under the new SCC:

```
# Replace <domain> with your DocumentProcessor domain name (for example, deucalion-domain)
oc delete pod -n ibm-cas \
  -l app.kubernetes.io/component=cast-runtime,app.kubernetes.io/part-of=<domain>
oc delete pod -n ibm-cas -l app.kubernetes.io/component=vllm
oc delete pod -n ibm-cas -l app.kubernetes.io/name=docling
```

Result

After the pods restart, confirm that they are admitted under the new SCC:

```
oc get pod <pod-name> -n ibm-cas \
  -o jsonpath='{.metadata.annotations.openshift\.io/scc}{"\n"}'
# Expected: ibm-cas-cast-runtime-mustrunasgroup
```

Upgrade stalls: Kafka CR is patched to an unsupported version

Problem statement

After enabling the cache filesystem or triggering any upgrade, the CAS operator logs repeat indefinitely:

```
Kafka not yet ready.
```

Inspecting the Kafka custom resource (CR) shows:

```
oc get kafka kafka -n ibm-cas -o jsonpath='{.status.conditions}'
[{
  "message": "Unsupported Kafka.spec.kafka.version: 3.9.0. Supported versions are: [4.0.0, 4.1.0]",
  "reason": "UnsupportedKafkaVersionException",
  "status": "True",
  "type": "NotReady"
}]
```

Important: This issue is recurrent. Re-apply the fix each time the CAS operator patches the Kafka CR to an unsupported version, that is, on every install re-entry or upgrade that triggers `UpdateKafkaVersion`.

Cause

The CAS operator hardcodes Kafka version 3.9.0 when patching the Kafka CR during upgrades and install re-entry. AMQ Streams 3.1.0-14 supports only Kafka versions 4.0.0 and 4.1.0. As a result, the Kafka CR transitions to a **`NotReady`** state immediately after the update, and the **`IsHealthyKafkaCluster`** check never returns **`true`**.

When to apply

Apply this fix before triggering install re-entry, that is, before applying the **`CasInstall`** status patch that re-enters the cache filesystem install path. Strimzi requires approximately 30 seconds to reconcile the Kafka CR to **`Ready`** after this patch.

If the operator has already entered the loop, apply the patch and wait for the next reconcile cycle to proceed.

Resolution

Patch the Kafka CR to set a supported version:

```
oc patch kafka kafka -n ibm-cas --type=merge \
  -p '{"spec":{"kafka":{"version":"4.0.0"}}}'
```

Result

Within approximately 30 seconds, the Kafka CR shows **`type=Ready`**, **`status=True`**.

Verify by running the following command:

```
oc get kafka kafka -n ibm-cas \
  -o jsonpath='{.status.conditions[?(@.type=="Ready")].status}{"\n"}'
# Expected: True
```

The operator log stops repeating `Kafka not yet ready` after the next reconcile cycle.

CAS installation fails with CAS Cache Filesystem Failed error

Problem statement

When installing CAS with Local Data Caching for Content Aware Storage (CAS) enabled, the installation stops and eventually fails with a message similar to:

```
CAS Cache Filesystem Failed
```

The installation does not progress past the cache filesystem validation stage.

Cause

When Local Data Caching for Content Aware Storage (CAS) is enabled, CAS waits for the IBM Storage Scale cache filesystem (**`cache-fs`**) to reach a healthy state before continuing the installation.

By default, CAS waits up to 20 minutes for the cache filesystem to become healthy. If the filesystem does not reach a healthy state within this period, the installation is canceled and marked as failed.

This situation can occur when:

- Filesystem creation is taking longer than expected.
- The **`cache-fs`** filesystem requires additional time to become healthy.

Resolution

If additional time is required for the cache filesystem to become healthy, configure the CAS installation to ignore the filesystem health timeout.

1. Verify the CAS installation resource:

   ```
   oc get casinstall -n ibm-cas
   ```

2. Patch the CAS installation resource to add the following annotation:

   ```
   oc patch casinstall ibm-cas-service-instance \
     -n ibm-cas \
     --type merge \
     -p '{"metadata":{"annotations":{"ignore-filesystem-health-timeout":"true"}}}'
   ```

3. Verify that the annotation is applied:

   ```
   oc get casinstall ibm-cas-service-instance \
     -n ibm-cas \
     -o yaml
   ```

Result

After you apply the annotation, CAS overrides the default 20-minute cache filesystem health timeout. The installation continues to wait until the IBM Storage Scale cache filesystem becomes healthy, and then proceeds.

Note: This annotation only affects installations that use Local Data Caching for Content Aware Storage (CAS).
Important: Use this workaround only when the cache filesystem is expected to become healthy eventually.

EventManager NetworkPolicy blocks CAS operator alert POSTs

Problem statement

The CAS operator (**`ibm-isf-cas-operator-controller-manager`**) calls **`PostInfo()`** and **`PostAlert()`** from the **`isf-common`** event client library during every **`setInstallationStageAndStatus`** reconcile step. These calls POST to https://<clusterIP>:10666/api/v1/eventmanager/alerts. As a result, no CAS events are reported in the IBM Fusion UI events section.

Cause

The **`eventmanager-network-policy`** in **`ibm-spectrum-fusion-ns`** has no ingress rule permitting traffic from **`ibm-cas`**. Hence, every outbound POST is silently dropped. The HTTP client also lacks a short timeout, so each POST blocks for the full OS or kernel TCP timeout before returning **`context deadline exceeded`**.

Resolution

Apply the following custom NetworkPolicy in **ibm-spectrum-fusion-ns** to grant the CAS operator pod ingress to eventmanager on port **10666**.

1. Create a YAML file with the following content:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-cas-operator-to-eventmanager
  namespace: ibm-spectrum-fusion-ns
  labels:
    app.kubernetes.io/managed-by: cas-operator
spec:
  podSelector:
    matchLabels:
      app: isfemdep
  policyTypes:
  - Ingress
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          kubernetes.io/metadata.name: ibm-cas
      podSelector:
        matchLabels:
          app.kubernetes.io/component: operator
          app.kubernetes.io/name: cas.isf.ibm.com
    ports:
    - port: 10666
      protocol: TCP
```

2. Apply with the following command:

```
oc apply -f allow-cas-operator-to-eventmanager.yaml
```

Important: This workaround is not durable. The **isf-serviceability-operator** (label **app.kubernetes.io/managed-by: isf-serviceability-operator**) manages **eventmanager-network-policy**, which might reconcile over or conflict with the standalone policy. The custom policy is also not applied automatically on fresh installations.

PVC remains in Pending state

Problem statement

The data source remains in the Connecting state and the associated PersistentVolumeClaim (PVC) stays in the **Pending** state.

The PVC events might contain an error similar to the following:

```
rpc error: code = Internal desc = failed to update comment for fileset [...]
rpc error: code = Internal desc = remote call failed with response
400 Invalid value in 'filesetName'
```

Cause

This issue can occur when the NFS export path that is specified during data source creation is no longer valid or accessible.

Resolution

1. Verify that the configured NFS export path still exists on the NFS server.
2. If the export path is changed:
   a. Delete the data source that references the outdated NFS export path.
   b. Obtain the current NFS export path from the NFS server.
   c. Create a new data source by using the updated NFS export path.

After the data source is re-created with a valid export path, the PVC must progress from the **Pending** state and the data source must eventually transition to the Connected state.

PVC remains in Pending state because the volume cannot be provisioned

Problem statement

The data source remains in the Connecting state and the associated PersistentVolumeClaim (PVC) stays in the **Pending** state.

Review the PVC events:

1. Open the data source PVC.
2. Select the Events tab.
3. Look for messages similar to the following:

```
Waiting for a volume to be created either by the external provisioner 'spectrumscale.csi.ibm.com' or manually by the
system administrator. If volume creation is delayed, please verify that the provisioner is running and correctly
registered.
```

Cause

The prior message indicates that IBM Storage Scale has not yet provisioned the volume that is requested by the PVC.

Possible causes include:

- The IBM Storage Scale cluster is running low on available capacity.
- The IBM Storage Scale cluster is temporarily unable to create new filesets or volumes.
- Volume provisioning is taking longer than expected due to cluster activity or resource constraints.

Resolution

1. Verify that the IBM Storage Scale cluster has sufficient free capacity.
2. If the cluster is full or nearly full, free up storage space and wait for provisioning to resume.
3. Monitor the PVC status and events for updates.
4. Wait for the PVC to transition from **`Pending`** to **`Bound`**.

Note: By default, an NFS data source can take up to one hour to transition to the Connected state. If the PVC is still progressing and no provisioning errors are reported, allow sufficient time for the provisioning process to complete before taking corrective action.
After the PVC reaches the **`Bound`** state, the data source must continue the connection process and eventually transition to the Connected state.

IBM Fusion or Global Data Platform upgrade gets stuck due to ibm-cas/kafka-broker-nodepool eviction

Problem statement

Global Data Platform upgrade gets stuck in between as ibm-cas/kafka-broker-nodepool-0 fails to evict.

Cause #1

The nodepools have anti-affinity configured correctly, but the eviction logic is restrictive. After one pod is evicted, the system does not allow any additional pods to be taken down, which prevents further rescheduling.

Cause #2

IBM Storage Scale includes a Pod Disruption Budget (PDB) for its node pods with a similar restrictiveness to the Kafka pods. Sometimes, a conflict can occur between the Kafka PDB and the IBM Storage Scale PDB. When this happens, the **`scale-core`** pod might remain in the **`Terminating`** state, and block the rolling update of a Machine Config Pool (MCP).

Resolution

1. Update Kafka custom resource (CR)
   a. In the OpenShift® console, go to Installed Operators > Streams for Apache Kafka (ibm-cas ns) > All Instances > Kafka > YAML.
   b. Locate the **`Kafka`** section and add:

   ```
   template:
     podDisruptionBudget:
       maxUnavailable: 2
   ```

   c. Save and apply the updated **`Kafka`** CR.
2. If one or more relevant scale-core pods are stuck in **`Terminating`** state for an extended period, follow these steps:
   a. Log in to the OpenShift cluster and check the pod status in the **`openshift-machine-config-operator`** namespace:

   ```
   oc get pods -n openshift-machine-config-operator
   ```

   b. Identify the **`machine-config-controller`** pod (for example: **`machine-config-controller-669b794b8f-6p4bl`**).
   c. View the logs to determine where MCP is blocked:

   ```
   oc logs machine-config-controller-669b794b8f-6p4bl -n openshift-machine-config-operator
   ```

   d. If the logs indicate that a pod is blocking MCP, manually evict that pod from the node:

   ```
   oc adm drain compute-1-ru4.racke.mydomain.com --force=true --ignore-daemonsets=true --grace-period=-1 --delete-emptydir-data
   ```

   e. Delete the **`scale-core`** pod:

   ```
   kubectl delete pod <scale-core-pod-name> --grace-period=0 --force
   ```

Strimzi migration from ZooKeeper to KRaft gets stuck at early CAS upgrade stage

Problem statement

The Strimzi cluster operator pod gets stuck in a no-op reconciliation loop. The following **`cas-operator`** log message reconciles multiple times:

```
2025-10-30T00:49:10Z  INFO  KRaft migration in progress... requeueing    {"controller": "casinstall", "controllerGroup": "cas.isf.ibm.com", "controllerKind": "CasInstall", "CasInstall": {"name":"ibm-cas-service-instance","namespace":"ibm-cas"}, "namespace": "ibm-cas", "name": "ibm-cas-service-instance", "reconcileID": "07107592-ca70-4d44-9a96-b66fa29f20af"}
```

If the metadata state of the Kafka Custom Resource (CR) is in **`KRaftMigration`** for a long time, it means that it is stuck:

```
% oc get Kafka
NAME  DESIRED KAFKA REPLICAS  DESIRED ZK REPLICAS  READY  METADATA STATE  WARNINGS
kafka  3            3           True  KRaftMigration
```

`amq-streams-cluster-operator` logs:

```
2025-10-30 01:03:33 INFO StrimziPodSetController:349 - Reconciliation #632(watch) StrimziPodSet(ibm-cas/kafka-broker-nodepool): StrimziPodSet will be reconciled
2025-10-30 01:03:33 INFO StrimziPodSetController:385 - Reconciliation #632(watch) StrimziPodSet(ibm-cas/kafka-broker-nodepool): reconciled
2025-10-30 01:03:33 INFO StrimziPodSetController:349 - Reconciliation #633(watch) StrimziPodSet(ibm-cas/kafka-broker-nodepool): StrimziPodSet will be reconciled
2025-10-30 01:03:33 INFO StrimziPodSetController:385 - Reconciliation #633(watch) StrimziPodSet(ibm-cas/kafka-broker-nodepool): reconciled
2025-10-30 01:03:33 INFO StrimziPodSetController:349 - Reconciliation #634(watch) StrimziPodSet(ibm-cas/kafka-zookeeper): StrimziPodSet will be reconciled
2025-10-30 01:03:33 INFO StrimziPodSetController:385 - Reconciliation #634(watch) StrimziPodSet(ibm-cas/kafka-zookeeper): reconciled
2025-10-30 01:03:33 INFO StrimziPodSetController:349 - Reconciliation #635(watch) StrimziPodSet(ibm-cas/kafka-controller-nodepool): StrimziPodSet will be reconciled
```

```
2025-10-30 01:03:33 INFO StrimziPodSetController:385 - Reconciliation #635(watch) StrimziPodSet(ibm-cas/kafka-controller-
nodepool): reconciled
2025-10-30 01:03:35 INFO ClusterOperator:140 - Triggering periodic reconciliation for namespace ibm-cas
2025-10-30 01:03:35 INFO AbstractOperator:266 - Reconciliation #636(timer) Kafka(ibm-cas/kafka): Kafka kafka will be
checked for creation or modification
2025-10-30 01:03:38 INFO AbstractOperator:546 - Reconciliation #636(timer) Kafka(ibm-cas/kafka): reconciled
2025-10-30 01:05:35 INFO ClusterOperator:140 - Triggering periodic reconciliation for namespace ibm-cas
2025-10-30 01:05:35 INFO AbstractOperator:266 - Reconciliation #637(timer) Kafka(ibm-cas/kafka): Kafka kafka will be
checked for creation or modification
2025-10-30 01:05:38 INFO AbstractOperator:546 - Reconciliation #637(timer) Kafka(ibm-cas/kafka): reconciled
```

Resolution

Restart the **amq-streams-cluster-operator** pod. This resumes the migration process.

Errors in CAS runtime pod after Scale upgrade

Problem statement

Data ingestion fails with errors in CAS runtime pod.

Cause #1

The filesystem might be unmounted. This error can be observed in the **cas-runtime** logs as a **NodeNotReadyError**, or a pod is stuck at **Init:0/1** or in a **CrashLoopBackOff** state with a **FailedMount** event.

Resolution

1. Ensure that the filesystem is mounted correctly on all required nodes where Kafka pods and other affected pods are running.
2. To check which nodes the filesystem is mounted on, go to the Scale cluster and run the following command:

   ```
   mmlsmount all -L
   ```

   This lists all the nodes that filesystem is mounted on.

3. On the OpenShift Container Platform cluster, in the **ibm-spectrum-scale** namespace, identify the corresponding pod.
4. In the **ibm-spectrum-scale** namespace, find a healthy pod that is one where the filesystem is correctly mounted.
5. Run the following command to mount the filesystem on all the necessary pods:

   ```
   mmmount all -a
   ```

   It also resolves any filesystem mount-related errors.

6. Restart all the Kafka pods, **cas-runtime** pod, and **nv-ingest** pods.

Cause #2

The **nv-ingest** job fails.

Resolution

1. Check whether the NV-ingest Redis pod is restarted. It can cause **nv-ingest** job to fail.
2. Restart the pods corresponding to **nv-ingest**.

Cause #3

The Kafka pods fail.

Resolution

1. This issue can be related to the filesystem mount problem described in Cause #1. Follow the resolution steps in Cause #1 if the error is related to filesystem mount.
2. Restart the Kafka pods.

Inaccurate search results using the Docling Multimodal Processing Engine

Problem statement

Intermittent issues are observed where search results from a domain using Docling Multimodal Processing Engine return random results.

Cause

The vector index becomes corrupted and results in inaccurate search results.

Resolution

To fix the issue, drop and recreate the vector index on the affected domain by following these steps:

1. Log in to the OpenShift cluster by running the following command through command line:

   ```
   oc login <cluster-url>
   ```

2. Switch to the **ibm-cas** project:

   ```
   oc project ibm-cas
   ```

3. Save the database password for later use:

   ```
   oc get secret cluster-parade-app -oyaml -n ibm-cas | yq '.data.password' | base64 -d
   ```

4. Access the parade database pod:

   ```
   oc exec -it cluster-parade-1 -- psql -U app -d app -h localhost -W
   ```

   Enter the database password retrieved in Step 3.

5. Run the following command to view table details:

   ```
   \d nameOfDomain
   ```

   Locate the index labeled `diskann (vector)`.

   Example output:

   ```
   app=> \d red1
                                      Table "public.red1"
         Column       |        Type         | Collation | Nullable |              Default
   -------------------+---------------------+-----------+----------+-----------------------------------
    pk                | bigint              |           | not null | nextval('red1_pk_seq'::regclass)
    fileid            | character varying   |           |          |
    text              | character varying   |           |          |
    vector            | vector(768)         |           |          |
    source            | jsonb               |           |          |
    content_metadata  | jsonb               |           |          |
   Indexes:
       "red1_pkey" PRIMARY KEY, btree (pk)
       "red1_file_idx" btree (fileid)
       "red1_vector_idx" diskann (vector)
   ```

6. Drop the `diskann (vector)` index:

   ```
   DROP INDEX nameOfDiskAnnIndex
   ```

   Example:

   ```
   app=> DROP INDEX red1_vector_idx;
   DROP INDEX
   ```

7. Verify the table details and confirm that the vector index no longer appears:

   ```
   \d nameOfTable
   ```

8. Recreate the vector index with the following command:

   ```
   CREATE INDEX nameOfDomain_vector_idx on public.nameOfDomain USING diskann (vector vector_cosine_ops);
   ```

   Example:

   ```
   app=> CREATE INDEX red1_vector_idx on public.red1 USING diskann (vector vector_cosine_ops);
   NOTICE:  Starting index build with num_neighbors=-1, search_list_size=100, max_alpha=1.2,
   storage_layout=SbqCompression.
   WARNING:  Indexed 96 tuples
   CREATE INDEX
   ```

9. Verify the table details and confirm that the new vector index appears in the list:

   ```
   \d nameOfTable
   ```

   Example:

   ```
   app=> \d red1
                                      Table "public.red1"
         Column       |        Type         | Collation | Nullable |              Default
   -------------------+---------------------+-----------+----------+-----------------------------------
    pk                | bigint              |           | not null | nextval('red1_pk_seq'::regclass)
    fileid            | character varying   |           |          |
    text              | character varying   |           |          |
    vector            | vector(768)         |           |          |
    source            | jsonb               |           |          |
    content_metadata  | jsonb               |           |          |
   Indexes:
       "red1_pkey" PRIMARY KEY, btree (pk)
       "red1_file_idx" btree (fileid)
       "red1_vector_idx" diskann (vector)
   ```

Errors in CAS operator after upgrading to CAS v1.0.6

Problem statement

If IBM Data Cataloging is installed, errors can be observed in the CAS operator:

```
`2025-10-02T15:17:35Z    ERROR   Failed to get DCS Instance`
```

Cause

If IBM Data Cataloging is installed by using Fusion GUI but it is not enabled for filtering, CAS operator repeatedly shows an error when it looks up the IBM Data Cataloging instance. This issue does not happen if you install IBM Data Cataloging through CLI.

Resolution

After you upgrade CAS to version 1.0.6, if IBM Data Cataloging is installed but you do not plan to enable filtering for CAS, follow these steps:

1. Scale down the CAS operator:

   ```
   oc scale deployment ibm-isf-cas-operator-controller-manager --replicas=0 -n ibm-cas
   ```

2. Patch the `CasInstall` custom resource (CR):

   ```
   oc patch casinstall ibm-cas-service-instance -n ibm-cas --type=merge -p '{"spec":{"enabled_features":{"dcs":
   {"enable":false,"instance_name":"data-cataloging-service-instance","namespace":"ibm-data-cataloging"}}}}'
   ```

3. Scale up the CAS operator:

```
oc scale deployment ibm-isf-cas-operator-controller-manager --replicas=1 -n ibm-cas
```

Errors in existing document processor after upgrading to CAS v1.0.6

Problem statement

After upgrading CAS from v1.0.5 to v1.0.6, errors can be observed in ingestion in the document processor logs. If you are running one or more Docling Multimodal pipelines, the following error can be observed in the document processor (`cast-runtime`) pod logs:

```
`2025-10-02T07:05:26Z - xxx - ERROR - xxx - Repo id must be in the form 'repo_name' or 'namespace/repo_name': '/models/granite-embedding-278m-multilingual'. Use `repo_type` argument if needed.`
```

Cause

With CAS 1.0.6 release, the deployment `spec` for the Docling Multimodal document processors (`cast-runtime`) is changed to mount a PVC with the required LLM models. You need to force the operator to deploy a new deployment for each document processor that is running `docling_multimodal` type pipeline.

Resolution

After you upgrade CAS to version 1.0.6, redeploy all the document processors that are running the `docling_multimodal` type pipeline:

1. Scale down the CAS operator.

   ```
   oc scale deployment ibm-isf-cas-operator-controller-manager --replicas=0 -n ibm-cas
   ```

2. Identify `DocumentProcessor` CRs that are running the `docling_multimodal` type pipeline.

   ```
   oc get DocumentProcessor -n ibm-cas
   ```

   Alternatively, you can use `jq` to filter by type.

   ```
   oc get documentprocessor -ojson| jq '.items[] | select(.spec.type == "docling_multimodal") | .metadata.name'
   ```

3. Delete all the deployments associated with those CRs.

   ```
   oc delete deployment <DocumentProcessor CR name>  -n ibm-cas
   ```

4. Scale up the CAS operator.

   ```
   oc scale deployment ibm-isf-cas-operator-controller-manager --replicas=1 -n ibm-cas
   ```

Data Cataloging `SpectrumDiscover` CR fails to display `SuccessfullyEnabled` status

Problem statement

When you update `CasInstall` Custom Resource (CR) to enable filtering by Data Cataloging, `SpectrumDiscover` CR fails to display `SuccessfullyEnabled` status for `ContentAwareStorage` feature.

Cause

Run the following command and check if `isd-metadata-extender` pod is having any issue to start:

```
oc get pod -n ibm-data-cataloging
```

Resolution

To identify the issue that is preventing the pod to start, run the following command:

```
oc describe pod <isd-metadata-extender pod name>
```

No data ingested after domain creation with Data Cataloging filtering enabled

Problem statement

Data Cataloging is successfully enabled for filtering. When you create a domain by using a data source through UI (or a domain and document processor through CLI), the files that meet the filtering criteria are not ingested into the vector database as expected.

The `isd-metadata-extender` pod logs under the `ibm-data-cataloging` namespace shows the following error:

```
{"level":"error","ts":"2025-09-27T06:56:58Z","caller":"watch/document_processor.go:88","msg":"Error running scale partial scan","error":"unexpected status code 404 while running partial scan on connection fs1","stacktrace":"github.ibm.com/ProjectAbell/dcs-metadata-extender/pkg/watch.TriggerPartialScans\n\t/tmp/app/pkg/watch/document_processor.go:88"}
```

Cause

A connection to the Scale filesystem is not created.

Resolution

1. Go to the Data Cataloging UI and create a connection for the Scale filesystem.
2. Delete the domain and document processor CR and redeploy them.

No data ingested for any domain

Problem statement

No indication of data ingestion in logs of CAS runtime pods for any of the corresponding `DocumentProcessor`.

Cause

One possible area to check is the health of `cas-core` pod. Check whether it is up and running with the following command:

```
oc get pod -n ibm-cas
```

Resolution

Restart the `cas-core` pod or determine the issue by running the following command:

```
oc describe pod <cas-core pod name>
```

Error in `query_service` configmap and deployment

Problem statement

When you upgrade CAS from v1.0.3 to v1.0.4, a ConfigMap and some deployment configs must be added to prevent errors in Access Control List (ACL) processing.

Cause

Query service API fails to work when an external Identity Provider (IDP) token is provided.

Resolution

1. Create the following ConfigMap:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: query-search-config
  namespace: ibm-cas
data:
  use_self_cert: 'false'
```

2. Update the query search deployment:
   a. Set the `readOnlyRootFilesystem` attribute to 'true'.
   b. In the query service deployment's `env` section, add the following entry:

```
- name: ENABLE_FILE_LEVEL_SECURITY
  value: false
```

   c. In the `volumes` section, add the following section:

```
- name: query-search-config
  configMap:
    name: query-search-config
    defaultMode: 420
- name: ibm-cas-cabundle
  configMap:
    name: openshift-service-ca.crt
    items:
    - key: service-ca.crt
      path: service-ca.crt
```

   d. In the `volumeMounts` section, add the following section:

```
- name: "query-search-config"
  readOnly: true
  mountPath: "/etc/config/cm/"
- name: ibm-cas-cabundle
  readOnly: true
  mountPath: /etc/config/cabundle
```

The nv-ingest pod is in `imagepullback` error

Problem statement

The `nv-ingest` pod is in `imagepullback` error state due to too many requests for `docker.io`.

Cause and resolution

1. **Cause**:
   Too many requests for `docker.io`.

   Resolution: Authenticate to `docker.io`.

2. **Cause**:
   The `nv-ingest` pod is not able to pull the image correctly.

   Resolution: Check NVIDIA guide to fix the issue.

CAS ingestion fails

Problem statement

The CAS ingestion fails with the following error in the logs of the `cast-runtime` pods:

```
2025-03-26 22:46:36,725 - ERROR - Error during fetching, retrying...
Error: HTTPSConnectionPool(host='nv-ingest.nv-ingest.svc.cluster.local', port=7670):
Max retries exceeded with url: /v1/fetch_job/ (Caused by SSLError(SSLError(1, '[SSL] record layer failure (_ssl.c:1006)')))
```

Cause

It implies that the `https` NVIDIA NIM service is not available.

Resolution

1. Go to `cast-runtime` deployment.
2. In the value parameter, change `https` to `http`:

```
- name: NVMM_NIM_SERVICE
value: https://nv-ingest.nv-ingest.svc.cluster.local
```

to

```
- name: NVMM_NIM_SERVICE
value: http://nv-ingest.nv-ingest.svc.cluster.local
```

Error in semantic_search

Problem statement

In the query search pod, you can get the following errors during a **semantic_search**:

```
ERROR: querysearch/semantic_search failed Connection error.
Traceback (most recent call last):
File "/opt/app-root/lib64/python3.11/site-packages/httpx/_transports/default.py", line 101, in map_httpcore_exceptions
yield
File "/opt/app-root/lib64/python3.11/site-packages/httpx/_transports/default.py", line 250, in handle_request
resp = self._pool.handle_request(req)
File "/opt/app-root/lib64/python3.11/site-packages/httpcore/_backends/sync.py", line 154, in start_tls
with map_exceptions(exc_map):
File "/usr/lib64/python3.11/contextlib.py", line 158, in __exit__
self.gen.throw(typ, value, traceback)
File "/opt/app-root/lib64/python3.11/site-packages/httpcore/_exceptions.py", line 14, in map_exceptions
raise to_exc(exc) from exc
```

Cause

It implies that the TLS version of the NVIDIA service is not available.

Resolution

1. Scale down the query-search deployment.
2. In the query-search deployment, change the value of variable:

   ```
   - name: NVMM_EMBED_SERVICE
   value: 'https://llama-nemotron-embed-1b-v2.nv-ingest.svc.cluster.local'
   ```

   to

   ```
   - name: NVMM_EMBED_SERVICE
   value: 'http://llama-nemotron-embed-1b-v2.nv-ingest.svc.cluster.local'
   ```

3. If the NVMM Nemo Ranking service is enabled in CAS Install CR, change the following value in the query-search deployment. Change the value of variable:

   ```
   - name: NVMM_NEMO_RANKER_SERVICE
   value: 'https://llama-nemotron-rerank-1b-v2.nv-ingest.svc.cluster.local'
   ```

   to

   ```
   - name: NVMM_NEMO_RANKER_SERVICE
   value: 'http://llama-nemotron-rerank-1b-v2.nv-ingest.svc.cluster.local'
   ```

4. Scale the deployment back up to its original number.

Datasource stuck in "Connecting" state error

Problem statement

The create Datasource may get stuck in a "connecting" state due to different reasons. To do a proper diagnostic, check the CAS operator logs to find the error message.

Cause and resolution

The possible causes of this error are as follows:

1. **Cause:**
   Scale CSI user does not have enough privileges for Watcher creation.

   ```
   ERROR RETURNED BY sConn.CreateWatch4Fileset ==========>
           =====> [EFSSG0012C Permission denied: Your role(s): [csiadmin, containeroperator], required
           role(s): [admin, storageadmin, securityadmin]] Unable to enable clustered watch for source
           castFS:root
   ```

   Resolution:

   Add the Scale CSI user to the Storage Administrator group in Scale as specified Configuring Scale user to enable watch creation.

2. **Cause:**
   Kafka authentication ConfigMap is missing.

   ```
   2025-03-28T18:21:10Z ERROR Reconciler error {"controller": "datasource", "controllerGroup": "cas.isf.ibm.com", "controlle
   ```

   Resolution:

   Create the ConfigMap for Kafka authentication.

3. **Cause:**
   Access is denied to static PV that is created by the Datasource. If a functional issue is detected regarding static PV accessibility, it could be related to an access denied error. To validate, access the mount path by using the pod generated by the DocumentProcessor. The pod has the same name as the DocumentProcessor.

```
(app-root) sh-5.1$ cd /gpfs/gpfs3/fileset_sample
bash: cd: /app-root: Permission denied
```

Resolution:

Add a known GID to the Fileset in Scale and add an annotation to the Datasource in CAS as described in Step 8.

Known limitations

- When you open the data source Details page from the IBM Fusion UI, an error message about failing to fetch data source insights might appear. If IBM Data Cataloging is not installed on the cluster, you can ignore this message. The error is relevant only when IBM Data Cataloging is installed.
- When you deploy the `CasInstall` version 1.1.0 with `docling_multimodal enabled: true`, the `vllm-vision` pod results in `CrashLoopBackoff` status. The pod and its status can be ignored as the failing pod causes no functional impact.
- Deploying an S3 data source with a custom SSL certificate is not supported for Filesystem with IBM Storage Scale version 5.2.x.
- When you upgrade CAS from v1.0.5 to v1.0.6, two PVCs are orphaned and must be manually deleted to reclaim storage. To delete these PVCs, run the following commands:

  ```
  oc delete pvc vllm-embedding-pvc
  oc delete pvc vllm-vision-pvc
  ```

- When you create a domain in the user interface, the `docling_multimodal` pipeline is automatically enabled. To disable it, update the `DocumentProcessor` custom resource (CR) and set the `dataType` of the new domain to `nvidia_multimodal`.
- Support for CAS on Azure Red Hat OpenShift (ARO) has additional licensing requirements from Red Hat due to the need to change the maximum number of threads for the pods. For more information, see the Red Hat knowledge base.
- Each domain must use a unique datasource location, that is, unique Scale path or unique S3 location. CAS currently does not support sharing same datasource location among multiple domains.

Related information

IBM publications, IBM MediaCenter videos, and external videos that contain information that is related to Content-Aware Storage (CAS).

IBM publications

- Content-Aware IBM Storage Scale
- Accelerating AI with IBM Storage Scale and NVIDIA

IBM MediaCenter videos

- IBM Content Aware Storage
- Content Aware Storage (CAS) with IBM Fusion
- AI Data Assistance with IBM Content Aware Storage (CAS)

External videos

- What is Content-Aware Storage? Unlocking AI's Potential with RAG
- Content-Aware Storage: Powering AI Agents & Assistants with RAG

Lakehouse-Table [Technology preview]

The Lakehouse-Table service is a Kubernetes-native solution for building and managing lakehouse infrastructure on IBM Fusion. It provides a simplified experience for administrators to provision and manage Data Lakehouse capabilities.

Important:
Lakehouse-Table service is a Technology Preview feature and is subject to Technology Preview support limitations. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information, see Technology Preview Features Support Scope.

The Lakehouse-Table service enables administrators to create a foundational lakehouse environment, including object storage integration, Iceberg catalog management, and integration with the IBM watsonx.data. This service is the entry point for building a complete data lakehouse experience on IBM Fusion.

Key features and benefits

- It provides a simplified experience for building and managing lakehouse infrastructure.
- It is tightly integrated with IBM watsonx.data, enabling seamless data management and analytics.
- It automates infrastructure provisioning by using operators, reducing manual configuration and minimizing errors.
- It provides a Kubernetes-native experience, enabling administrators to manage lakehouse infrastructure using familiar tools and workflows.

- **Installing Lakehouse-Table service**
  Initiate the installation of Lakehouse-Table service from the user interface of IBM Fusion.

- **Configuring Lakehouse-Table service**
  Use the instructions to configure the Lakehouse-Table service by creating an `OpenTableCatalog` resource and verifying its creation.
- **Troubleshooting issues and known limitations in Lakehouse-Table**
  The common troubleshooting steps and limitations in the Lakehouse-Table service.

Installing Lakehouse-Table service

Initiate the installation of Lakehouse-Table service from the user interface of IBM Fusion.

Before you begin

To ensure a successful installation of the Software Hub service, verify that the following prerequisites are met:

1. IBM Fusion is installed and configured.
2. The necessary storage classes are available:
   - Block storage (RBD)
   - Filesystem storage (CephFS)
   - Object storage (RGW)
3. Ensure that you apply the IBM Fusion service definition.

```
apiVersion: service.isf.ibm.com/v1
kind: FusionServiceDefinition
metadata:
  name: ibm-lakehouse-service
  namespace: ibm-spectrum-fusion-ns
spec:
  disableFIPSSupport: false
  hasRelatedDefinition: false
  createResourceConfigMap: false
  supportedPlatforms:
    - BareMetalHCI
    - BareMetal
    - BareMetalSDS
    - VSphere
  subService: false
  serviceability:
    logcollector:
      namespaces:
        - ibm-software-hub-operator
  autoDeploy: false
  serviceInformation:
    lastUpdated: 1733952411788
    vendor: IBM
    name: BMYWD0001
    dialogDescription: BMYWD0201
    version: 5.4.0
    icon:
'
hbWU9IkJ1aWxkIGljb24gaGVyZSIgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIiB4bWxuczp4bGluaz0iaHR0cDovL3d3dy53My5vcmcvMTk5O
S94bGluayIgdmlld0JveD0iMCAwIDMyIDMyIj4KICA8ZGVmcz4KICAgIDxzdHlsZT4KICAgICAgLmNscy0xIHsKICAgICAgICBtYXNrOiBlcmwoI21hc2spOwo
gICAgICB9CgogICAgICAuY2xzLTIgewogICAgICAgIGZpbGw6ICMwMDFkNmM7CiAgICAgIH0KCiAgICAgIC5jbHMtMyB7CiAgICAgICAgZmlsbDogdXJsKCNsa
W5lYXItZ3JhZGllbnQtMik7CiAgICAgIH0KCiAgICAgIC5jbHMtNCB7CiAgICAgICAgZmlsbDogdXJsKCNsaW5lYXItZ3JhZGllbnQpOwogICAgICB9CgogICA
gICAuY2xzLTUgewogICAgICAgIGZpbGw6ICNmZmY7CiAgICAgIH0KCiAgICAgIC5jbHMtNiB7CiAgICAgICAgZmlsbDogdXJsKCNCbHVlX0lCTV9JY29uX0dyY
WRpZW50KTsKICAgICAgfQogICAgPC9zdHlsZT4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0ibGluZWFyLWdyYWRpZW50IiB4MT0iLTM0ODEuMTUiIHkxPSItODA
xLjA4IiB4Mj0iLTM0ODEuMTUiIHkyPSItODA2LjA4IiBncmFkaWVudFRyYW5zZm9ybT0idHJhbnNsYXRlKDM0ODcuNjUgODMyLjU4KSIgZ3JhZGllbnRVbml0c
z0idXNlclNwYWNlT25Vc2UiPgogICAgICA8c3RvcCBvZmZzZXQ9IjAiIHN0b3AtY29sb3I9IiMwMDAiIHN0b3Atb3BhY2l0eT0iMCIvPgogICAgICA8c3RvcCB
vZmZzZXQ9Ii42OCIgc3RvcC1jb2xvcj0iIzAwMCIvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0ibGluZWFyLWdyYWRpZ
W50LTIiIHgxPSItNDE0Ni42NSIgeTE9Ijc2LjQyIiB4Mj0iLTQxNDYuNjUiIHkyPSI3MS40MiIgZ3JhZGllbnRUcmFuc2Zvcm09InRyYW5zbGF0ZSg0MTc1LjY
1IC01NS40MikiIHhsaW5rOmhyZWY9IiNsaW5lYXItZ3JhZGllbnQiLz4KICAgIDxtYXNrIGlkPSJtYXNrIiB4PSIwIiB5PSIwIiB3aWR0aD0iMzIiIGhlaWdod
D0iMzIiIG1hc2tVbml0cz0idXNlclNwYWNlT25Vc2UiPgogICAgICA8Zz4KICAgICAgICA8cGF0aCBjbGFzcz0iY2xzLTUiIGQ9Ik0yNy44NywySDQuMTNjLTE
uMTgsMC0yLjEzLjk1LTIuMTMsMi4xM3YyMy43NGMwLDEuMTguOTUsMi4xzLDIuMTNoMy44N3YtMmgtMy44N2MtLjA3LDAtLjEzLS4wNi0uMTMtLjEzV
jEwLjVoMjR2OS41aDJWNC4xM2MwLTEuMTgtLjk1LTIuMTMtMi4xMy0yLjEzWk0yOCw4LjVINHYtNC4zN2MwLS4wNy4wNi0uMTMuMTMtLjEzaDIzLjc0Yy4wNyw
wLC4xMy4wNi4xMy4xMy4xM3Y0LjM3WiIvPgogICAgICA8cGF0aCBjbGFzcz0iY2xzLTUiIGQ9Ik00IGNsYXNzPSJjbHMtNCIgeD0iMTcuNzUiIHk9IjEzLjc0I
mFuc2Zvcm09InRyYW5zbGF0ZSgzNS41IDIyLjUpIHJvdGF0ZSg5MCkiLz4KICAgICAgICA8cmVjdCBjbGFzcz0iY2xzLTMiIHg9IjIyIi4iB5PSIxNiIgd2lkdGg
9IjIiIGhlaWdodD0iNSIgdHJhbnNmb3JtPSJ0cmFuc2xhdGUoNTggMzcpIHJvdGF0ZSgxODApIi8+CiAgICAgICAgPHJlY3QgY2xzLTMiIHg9IjIyIi4iB5PSIxNiIgd2lkdGg
XJHcmFkaWVudCBpZD0iQmx1ZV9JQk1fSWNvbl9HcmFkaWVudCIgZGF0YS1uYW1lPSJCbHVlIElEdyYWRpZW50IiB4MT0iMCIgeTE9IjE2IiB4Mj0
iMzIiIHkyPSIxNiIgZ3JhZGllbnRVbml0cz0idXNlclNwYWNlT25Vc2UiPgogICAgICA8c3RvcCBvZmZzZXQ9Ii4xIiBzdG9wLWNvbG9yPSIjMDBiOGIyIi8+C
iAgICAgIDxzdG9wIG9mZnNldD0iLjkiIHN0b3AtY29sb3I9IiMwMDVhZmYiLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgPC9kZWZzPgogIDxnIGNsYXNzPSJ
jbHMtMSI+CiAgICA8cGF0aCBjbGFzcz0iY2xzLTYiIGQ9Ik0wLDMyVjBoMzJWMzJIMFoiLz4KICA8L2c+CiAgPHBhdGggc3M9ImNscy0yIiBkPSJNMjgsM
jJjLTEuMzktLjA4LTIuNi0xLTMuMDUtMi4zMi0uMTQtLjQtLjUyLS42OC0uOTUtLjY4cy0uOTUtLjI4LS42OC0uNDUtMS4zMi0xLjY2LDIuMjQtMy4wNSw
yLjMyLTEuMzktLjA4LTIuNi0xLTMuMDUtMi4zMi0uMTQtLjQtLjUyLS42OC0uOTUtLjY4LS40My0uMDEtLjgyLjI3LS45NS42OC0uNDUsMS4zMi0xLjY2LDIuM
jQtMy4wNSwyLjMyaC0xYy0uNTUsMC0xLC40NS0xLDFoMGMwLC41NS40NSwxLDEsMWgxYzEuNTcuMDEsMy4wNi0uNzIsNC0xLjk5LDEuNjcsMi4yMS0xLjE4LMWgtM
VoiLz4KICA8cGF0aCBjbGFzcz0iY2xzLTIiIGQ9Ik0yOCwyOGMtMS4zOS0uMDgtMi42LTEtMy4wNS0yLjMyLTEuMzktLjI4LTIuNi0xLTMuMDUtMi4zMi0uMTQtLjQtLjUyLS42OC0uOTUtLjY4cy0uOTUuMjgtLjY4LS40NS0uNjgtLjQzLS4wMS0uODIuMjctLjk1LjY4LS40NSwxLjMyLTEuNjYsMi4yMS0xLjY4LDMuMDUtMi4zMi4xNC0uNC41Mi0uOTUuNjgtLjk1cy45NS0uMjgsMS4zMi0uNjgtLjQzLS44Mi0xLjY2LTIuMjQtMy4wNSwyLjMyLTEuMzkuMDgtMi42LDEtMy4wNSwyLjMyLTEuMzkuMjgtMi42LDEtMy4wNSwyLjMyLTEuMzkuMDgtMi42LT
```

```
    category: BMYAI0202
  relatedDefinitionSpec: {}
  skipServiceDiscovery: false
  onboarding:
    isSingleton: true
    resourceKind: LakehouseManager
    syncServiceCRSpec: false
    groupName: lakehouse-table.fsh.ibm.com
    parameters:
      - dataType: string
        defaultValue: ibm-software-hub-operator
        descriptionCode: BMYSRV00003
        displayNameCode: BMYSRV00004
        name: namespace
        required: true
        userInterface: false
    multiVersionService: false
    serviceOperatorSubscription:
      triggerCatSrcCreate: false
      isClusterWide: false
      channel: v1.0
      skipDeployOperator: false
      readImageFromFSI: false
      catalogSourceDetails:
        displayName: Fusion Sofewarehub Catalog
        imageName: ibm-fsh-operator-catalog
        imageTag: v1.0
        publisher: IBM
        registryPath: icr.io/cpopen
        skipCatSrcCreation: false
      operatorName: ibm-fsh-operator
      globalCatalogSource: false
      createCatalogSource: true
      catalogSourceName: ibm-fsh-operator-catalog
    version: v1
    resourcePluralName: lakehousemanagers
  preferredStorageClassList:
    - ocs-storagecluster-ceph-rbd
    - ocs-storagecluster-cephfs
  serviceInstallPrerequisite:
    config: {}
    kubeobject:
      - StorageClass
    ocpVersionMaxReq: '4.21'
    ocpVersionMinReq: '4.14'
```

4. The following table outlines the compatibility matrix for the components involved, including the minimum supported versions.

   Table 1. Compatibility Matrix for Components

Component	Minimum supported version
IBM Cloud Paks for Data	5.4
watsonx.data	2.4
Data Lakehouse-Table	1.0.0
Fusion Data Foundation	4.21

   Note: The versions listed are the minimum supported versions. Using versions later than those listed is supported unless otherwise noted.

About this task

Use this procedure to install the Lakehouse-Table service from the Services page of the IBM Fusion user interface.

Procedure

1. Before installing Lakehouse-Table, ensure that IBM **`Cert Manager`** is not installed and that the **`ibm-cert-manager`** namespace does not exist. If either is present, uninstall IBM **`Cert Manager`** and delete the namespace.
   Note: As part of a fresh installation, the latest version of Red Hat® certificate Manager is installed with Software Hub.
2. In the IBM Fusion user interface, go to Services page.
3. In the Available section, click the Lakehouse-Table tile.
   It opens a Lakehouse-Table service page.
4. In the Lakehouse-Table page, go through the features and capabilities of the service and click Install.
   It opens a Lakehouse-Table install service page.
   The installation starts and a notification appears on the Services page. You can see the progress of the installation in the Services > Installed section. After the installation is completed successfully, you can see the status as Healthy. A success notification gets displayed. It disappears automatically after some time.

   From the ellipsis menu, you can download logs and view documentation.

5. Validate the installation.
   - Verify the Lakehouse-Table installation from the IBM Fusion user interface:
     After you enable the Lakehouse-Table service, you can view the service version and health status. From the ellipsis menu, you can download logs and view documentation. After you successfully collect the logs, a success notification is displayed. The notification disappears automatically after some time.
   - Verify the Lakehouse-Table installation from the OpenShift® Container Platform web console:
     - In the OpenShift console, go to Home > CustomResourceDefinition.
     - In the CustomResourceDefinition page, search and check for the **`LakehouseManager`** CR.
       Lakehouse-Table service also installs Software Hub in a separate namespace with the **`SoftwareHubManager`** custom resource named **`lakehouse-table-softwarehub`**.

- After you enable the Lakehouse-Table service, you can view the service version and health status. From the ellipsis menu, you can download logs and view documentation. After you successfully collect the logs, a success notification is displayed. The notification disappears automatically after some time.

Configuring Lakehouse-Table service

Use the instructions to configure the Lakehouse-Table service by creating an `OpenTableCatalog` resource and verifying its creation.

Before you begin

Ensure that you met all the prerequisites before you configure the Lakehouse-Table.

1. Ensure that the Security Token Service (STS) must be enabled on your Ceph RGW cluster for IAM role-based access.
   a. Run the following command to enable the Ceph Toolbox pod:

   ```
   oc patch storagecluster ocs-storagecluster -n openshift-storage \
     --type json \
     --patch '[{ "op": "replace", "path": "/spec/enableCephTools", "value": true }]'
   ```

   b. Wait for the Toolbox application to initialize.

   ```
   oc get pods -n openshift-storage | grep rook-ceph-tools
   ```

   Example output:

   ```
   rook-ceph-tools-xxxxx   1/1     Running   0          30s
   ```

   c. Run the following command to access the Toolbox pod:

   ```
   TOOLBOX_POD=$(oc get pods -n openshift-storage -l app=rook-ceph-tools -o jsonpath='{.items[0].metadata.name}')
   oc -n openshift-storage rsh $TOOLBOX_POD
   ```

   d. Run the following command to find the RGW client name:

   ```
   ceph auth list | grep "client.rgw"
   ```

   Inside the toolbox pod, it lists all Ceph auth clients to find your RGW client.
   Example output:

   ```
   client.rgw.ocs-storagecluster-cephobjectstore.a
   ```

   The client name is formatted as follows: `client.rgw.<Ceph object store name>.<instance ID>`.
   e. To enable STS for the RGW client, update the `CephObjectStore` configuration.
   - Run the following commands to replace `<cephobjectstore>` with the name of your `CephObjectStore` (for example, `ocs-storagecluster-cephobjectstore`) and `<instance-id>` with the instance identifier (for example, `a`).

     ```
     ceph config set client.rgw.<cephobjectstore>.<instance-id> rgw_s3_auth_use_sts true
     ceph config set client.rgw.<cephobjectstore>.<instance-id> rgw_sts_key "abcdefghijklmnop"
     ```

     For example:

     ```
     ceph config set client.rgw.ocs-storagecluster-cephobjectstore.a rgw_s3_auth_use_sts true
     ceph config set client.rgw.ocs-storagecluster-cephobjectstore.a rgw_sts_key "abcdefghijklmnop"
     ```

     Important: The `rgw_sts_key` must be a secure, random, alphanumeric string that is 16 characters in length. The example that is provided uses a placeholder value. Ensure that you generate a secure key for your configuration.
   f. Run the following command to verify STS Configuration:

   ```
   ceph config get client.rgw.<cephobjectstore>.<instance-id> rgw_s3_auth_use_sts
   ceph config get client.rgw.<cephobjectstore>.<instance-id> rgw_sts_key
   ```

   g. To apply the changes, exit the Toolbox pod and restart the RGW pods by running the following commands:

   ```
   exit  # Exit toolbox pod
   oc delete pods -n openshift-storage -l app=rook-ceph-rgw
   ```

   h. Wait for the RGW pods to restart:

   ```
   oc get pods -n openshift-storage -l app=rook-ceph-rgw
   ```

2. Configure the global configmap as follows:
   Important:
   - Before creating catalogs, configure the global configmap for the operator to provide default values. This way, you can create minimal catalog custom resources without duplicating the configuration.
   - The global configmap is automatically populated by IBM Fusion. If necessary, you can update the details as required.
   a. Run the following command to edit the global configmap:

   ```
   oc edit configmap ibm-isf-lakehouse-tables-operator-global-config -n lakehouse-tables-operator-system
   ```

   b. Configure the required settings.
   For example:

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: ibm-isf-lakehouse-tables-operator-global-config
   ```

```
  namespace: lakehouse-tables-operator-system
data:
  # RGW Configuration
  RGW_ADMIN_USER_SECRET_NAME: "rook-ceph-object-user-ocs-storagecluster-cephobjectstore-lakehouse-admin"
  RGW_NAMESPACE: "openshift-storage"
  RGW_TLS_DISABLED: "false"
  RGW_TLS_CERT_CONFIGMAP_REF: "openshift-service-ca.crt"
  RGW_TLS_KEY: "service-ca.crt"

  # MDS Configuration
  MDS_URI: "https://ibm-lh-lakehouse-mds-rest-svc.lakehouse-table-cpd-
instance.svc.cluster.local:8180/mds/metadata/v1"
  MDS_USERNAME: "cpadmin"
  MDS_PASSWORD_SECRET_NAME: "ibm-iam-bindinfo-platform-auth-idp-credentials"
  MDS_NAMESPACE: "lakehouse-table-cpd-instance"
  MDS_AUTH_INSTANCE_ID: "1767781376189241"  # Get from ibm-lh-lakehouse-config-cm
  AMS_URI: "https://internal-nginx-svc.lakehouse-table-cpd-instance.svc.cluster.local:12443"

  # MDS Token Configuration
  MDS_TOKEN_TYPE: "Bearer"
  MDS_TOKEN_ENDPOINT: "https://internal-nginx-svc.lakehouse-table-cpd-instance.svc.cluster.local:12443/icp4d-
api/v1/authorize"

  # MDS TLS Configuration
  MDS_TLS_DISABLED: "false"
  MDS_TLS_CERT_SECRET_REF: "ibm-lh-tls-secret"
  MDS_TLS_KEY: "cabundle.crt"
```

Procedure

To create an `OpenTableCatalog` resource, follow these steps:

a. You can create an `OpenTableCatalog` resource by using either a minimal catalog configuration or an explicit catalog configuration.

Minimal catalog configuration

i. Create a file named `my-catalog-minimal.yaml` with the following content:

```
apiVersion: lakehouse-tables.isf.ibm.com/v1alpha1
kind: OpenTableCatalog
metadata:
  name: my-iceberg-catalog
  namespace: lakehouse-catalog-admin
spec:
  catalog:
    tableFormat: "iceberg"
```

ii. Run the following command to apply the resource:

```
oc apply -f my-catalog-minimal.yaml
```

Explicit catalog configuration (optional)

If you prefer to specify all values explicitly or override the global configuration, create a file that is named `my-catalog.yaml` with the following content:

```
apiVersion: lakehouse-tables.isf.ibm.com/v1alpha1
kind: OpenTableCatalog
metadata:
  name: my-iceberg-catalog
  namespace: lakehouse-table-cpd-instance
spec:
  objectStore:
    adminUser:
      secretName: rook-ceph-object-user-ocs-rgw-rgw-user
      secretNamespace: openshift-storage
    tls:
      disabled: true
  bucketName: my-data-bucket
  iamResources:
    account:
      accountId: "RGW12345678901234567"
      accountName: "my-data-account"
      rootUserName: "my-account-root"
    userName: "my-data-user"
    roleName: "my-data-role"
    policyName: "my-data-policy"
  catalog:
    catalogName: "my_catalog"
    server:
      lakehouse:
        mdsUri: "https://ibm-lh-lakehouse-mds-rest-svc.lakehouse-table-cpd-
instance.svc.cluster.local:8180/mds/metadata/v1"
        userName: "cpadmin"
        passwordSecretRef:
          secretName: "ibm-iam-bindinfo-platform-auth-idp-credentials"
          secretNamespace: "lakehouse-table-cpd-instance"
        token:
          type: "Bearer"
          endpoint: "https://cpd-lakehouse-table-cpd-instance.<your-domain>/icp4d-api/v1/authorize"
      tls:
        disabled: true
    tableFormat: "iceberg"
```

b. Run the following command to apply the resource:

```
oc apply -f my-catalog.yaml
```

c. Run the following commands to verify whether the `OpenTableCatalog` resource is created successfully:

```bash
oc -n lakehouse-tables-operator-system get otc my-iceberg-catalog -o jsonpath='{.status}' | jq -r .
{
"accountId": "RGW59281713583741298",
"info": {
"amsUri": "<https://internal-nginx-svc.lakehouse-table-cpd-instance.svc.cluster.local:12443",>
"authSecretName": "my-iceberg-catalog-bearer-fetch-credentials", "authSecretNamespace": "lakehouse-table-cpd-instance",
"catalogName": "my_iceberg_catalog", "catalogStorageBackend": "my-data-bucket", "mdsUri": "<https://ibm-lh-lakehouse-mds-rest-svc.lakehouse-table-cpd-instance.svc.cluster.local:8180/mds/metadata/v1",> "objectStoreEnpoint": "<https://rook-ceph-rgw-ocs-storagecluster-cephobjectstore.openshift-storage.svc:443",> "userName": "my-iceberg-catalog-zen-admin"
},
"lastSyncTime": "2026-06-05T17:40:59Z",
"phase": "Ready"
}
```

The operator creates the following resources:

- S3 bucket: `my-data-bucket`
- IAM account with root user
- IAM user, role, and policy
- Iceberg catalog registered in watsonx.data
- Secret access with credentials: `my-iceberg-catalog-fetch-bearer-token`.

d. To use the `OpenTableCatalogUser`, you must first obtain an OAuth token. Follow the steps to retrieve the token:

i. Run the following command to obtain the authentication secret name:

```
AUTH_SECRET="$(oc get opentablecatalog "$CAT_NAME" -n "$CAT_NS" -o jsonpath='{.status.info.authSecretName}')"
```

ii. Run the following command to display all authentication details:

```
oc get secret "$AUTH_SECRET" -n "$CAT_NS" -o json | jq -r '.data | to_entries[] | "KEY=\(.key)\nVALUE=\
(.value|@base64d)\n---"'
```

From the output, identify the `username` and `password` values.

iii. Use the `curl` command to generate the OAuth token. Replace `<username>` and `<PASSWORD>` with the values obtained in the previous step:

```
curl -k -X POST '<https://cpd-lakehouse-table-cpd-instance.apps.datalake.qe.ceph.tuc.ibm.com/icp4d-api/v1/authorize'>
\
-H 'Content-Type: application/json' \
-d '{"password":"<PASSWORD>","username":"<username>"}'
```

iv. Run the following command to get the `OpenTableCatalogUser` name:

```
oc describe otc <catalog_name> -n <otc_namespace>
```

e. Create additional `OpenTableCatalogUser`, add the CRD to the configuration.

Minimal CRD (catalog use only)

```
apiVersion: lakehouse-tables.isf.ibm.com/v1alpha1
kind: OpenTableCatalogUser
metadata:
  name: analyst-user
  namespace: cpd-instance
spec:
  catalogUser:
    userName: analyst
  resource:
    catalogRef: opentablecatalog-sample
    permission: use
```

Minimal CRD with L3 data policies

```
apiVersion: lakehouse-tables.isf.ibm.com/v1alpha1
kind: OpenTableCatalogUser
metadata:
  name: invoice-user
  namespace: cpd-instance
spec:
  catalogUser:
    userName: invoice-user
  resource:
    catalogRef: opentablecatalog-sample
    permission: use
    dataPolicies:
      - scope: table
        target: sales.invoices
        permissions: ["read"]
```

f. Optional: Run the following command to delete the resource:

```
oc delete otc my-iceberg-catalog -n lakehouse-tables-operator-system
```

The data is retained even if you delete the resource.

Troubleshooting issues and known limitations in Lakehouse-Table

The common troubleshooting steps and limitations in the Lakehouse-Table service.

Catalog creation fails

Problem statement
: The `OpenTableCatalog` resource shows an error in its status, indicating a failure in catalog creation.

Cause
: The cause of the failure can be attributed to several factors, including:

- Incorrect RGW admin credentials.
- Network connectivity issues to RGW and watsonx.data.
- Invalid TLS certificates if TLS is enabled.
- Detailed error messages available in the operator logs.

Resolution
: To resolve the issue, follow these steps:

1. Verify that the RGW admin credentials are correct.
2. Check the network connectivity to RGW and watsonx.data.
3. Ensure that the TLS certificates are valid if TLS is enabled.
4. Check the operator logs for detailed error messages to diagnose the issue.

Cannot access catalog from Spark or Trino

Problem statement
: Authentication errors occur when querying the catalog from Spark or Trino.

Cause
: The cause of the authentication errors can be due to:

- Missing or incorrect bearer token secret.
- Expired bearer token.
- Vended credentials not enabled in the catalog configuration.
- Inaccessibility of the S3 endpoint from the query engine pods.

Resolution
: To resolve the issue, follow these steps:

1. Verify that the bearer token secret exists using the command `oc get secret <catalog-name>-fetch-bearer-token`.
2. Check the token expiration and refresh it if necessary.
3. Ensure that vended credentials are enabled in the catalog configuration.
4. Verify that the S3 endpoint is accessible from the query engine pods.

Out-of-Memory (OOM) issues

Problem statement
: Exceeding the specified limits in the catalog table configuration can result in OOM issues on IO engines, MDS, and other components.

Symptoms

- OOM errors occur on IO engines, MDS, or other components.
- System performance degrades or becomes unresponsive.

Cause:
: The following limits, when exceeded, can cause OOM issues:

- Total artifacts exceeding 900 in a single catalog.
- Total number of schemas exceeding 250.
- Total number of tables exceeding 1250.
- Number of rows exceeding 80,000 (with 5 columns) or 250,000 (with 5 columns) in a table.
- Number of `OpenTableCatalogUser` exceeding 250.
- Number of IO engines (Trino) exceeding 5 with 4G RAM.

Resolution
: Follow the steps to resolve the issue:

1. Ensure that the total number of artifacts, schemas, tables, and rows are within the specified limits.
2. Distribute data across multiple catalogs, schemas, or tables to avoid exceeding the specified limits.
3. Limit the number of IO engines (Trino) to 5 or fewer with 4G RAM, or increase the RAM allocation as needed.
4. Regularly monitor system performance and adjust configuration as needed to prevent OOM issues.

Known limitations

- When installing Lakehouse-Table service, the `ibm-cpd-ae-operator` pod may encounter an `ImagePullBackOff` error. As a resolution, run the following command to patch the `ibm-cpd-ae-operator` deployment:

```
oc patch deployment ibm-cpd-ae-operator -n cpd-operator \
  --type='json' \
  -p='[
    {
      "op": "replace",
      "path": "/spec/template/spec/containers/0/image",
```

```
      "value": "cp.stg.icr.io/cp/ibm-cpd-analyticsengine-
operator@sha256:7f9ceed629755a77a4443059fee31b677a3d9f8dcad82336892ee77fd252b83e"
    }
  ]'
```

Deploying IBM Fusion services using CLI

Use the instructions to deploy the IBM Fusion services using the command-line interface (CLI).

Procedure

1. Log in to the OpenShift® terminal using the following command.

   ```
   oc login --token=${TOKEN_VALUE} --server=${SERVER_NAME}
   ```

2. Follow the steps to deploy the IBM Fusion services:

   Global Data Platform

   Important: This procedure applies to configuring the Global Data Platform service on Red Hat® OpenShift Container Platform version 4.20 or earlier. On version 4.21 or later, the Global Data Platform service is not available as a separate service and must be configured as part of Fusion Data Foundation. For more information, see Fusion Data Foundation service with Global Data Platform.

   a. Create the YAML to deploy the Global Data Platform.

      ```
      cat << EOF > accept_license.yaml
      apiVersion: prereq.isf.ibm.com/v1
      kind: SpectrumFusion
      metadata:
        name: spectrumfusion
        namespace: ibm-spectrum-fusion-ns
      spec:
        GlobalDataPlatform:
          Enable: true
      EOF
      ```

   b. Run the following command to apply the YAML.

      ```
      oc apply -f accept_license.yaml
      ```

   c. Run the following command to check the Global Data Platform installation status.

      ```
      oc get spectrumfusion -n ibm-spectrum-fusion-ns -o json
      ```

      Sample output:

      ```
       "status": {
                      "GlobalDataPlatformStatus": {
                          "ServiceEnabled": true,
                          "health": "Healthy",
                          "installStatus": "Completed",
                          "installStatusMessage": "IBM Storage Scale installation is succeeded",
                          "installStatusMessageCode": "BMYSS0003",
                          "isCRCRendered": false,
                          "isDeployable": true,
                          "isMachineConfigRendered": false,
                          "isSupported": true,
                          "isUpgradeFailed": false,
                          "isUpgradeInProgress": false,
                          "maxAvailableVersion": "5.2.1.1",
                          "progressPercentage": 100,
                          "upgradeAvailable": false,
                          "version": "5.2.1.1"
                      }
      ```

   Data Foundation

   a. Create the YAML to deploy the Data Foundation.

      ```
      cat << EOF > deploy_odf.yaml
      apiVersion: service.isf.ibm.com/v1
      kind: FusionServiceInstance
      metadata:
        name:  odfmanager
        namespace: ibm-spectrum-fusion-ns
      spec:
        creator: User
        doInstall: true
        parameters:
        - name: namespace
          provided: false
          value: openshift-storage
        - name: creator
          provided: false
          value: Fusion
        - name: backingStorageType
          provided: true
          value: Local
        - name: autoUpgrade
          provided: false
          value: "true"
        - name: enableLVMStorage
      ```

```
    provided: true
    value: "false"
  serviceDefinition: data-foundation-service
  triggerUpdate: false
EOF
```

Note: In the CR, provide a supported backingStorageType based on your platform.
For more information about platform support, see *Platform support table in the About the task section* of Installing Fusion Data Foundation.

In the CR, the valid input for backingStorageType > Value are `Dynamic`, `Local`, or `External`. Ensure that you capitalize the first letter of the value.

b. Run the following command to apply the YAML.

```
oc apply -f deploy_odf.yaml
```

Sample output:

```
fusionserviceinstance.service.isf.ibm.com/odfmanager created
```

c. Run the following command to verify whether the Data Foundation service is installed.

```
oc get FusionServiceInstance odfmanager -o json -n ibm-spectrum-fusion-ns
```

Sample output:

```
"installStatus": {
            "progressPercentage": 50,
            "retryOnFailure": false,
            "status": "Installing"
        }
```

IBM Data Cataloging

a. Create the YAML to deploy the IBM Data Cataloging.

```
cat << EOF > deploy_datacataloging.yaml
apiVersion: service.isf.ibm.com/v1
kind: FusionServiceInstance
metadata:
  name:  data-cataloging-service-instance
  namespace: custom-ns
spec:
  parameters:
    - name: namespace
      provided: false
      value: custom-ibm-data-cataloging
    - name: rwx_storage_class
      provided: true
      value: ocs-storagecluster-cephfs
    - name: doInstall
      provided: true
      value: 'true'
    - name: license
      provided: true
      value: '{"accept": true}'
    - name: affinity
      provided: false
      value: '{"nodeAffinity":{"requiredDuringSchedulingIgnoredDuringExecution":{"nodeSelectorTerms":
[{"matchExpressions":[{"key":"nvidia.com/gpu","operator":"NotIn","values":["Exists"]},
{"key":"nvidia.com/gpu.present","operator":"NotIn","values":["true"]},
{"key":"gpu.isf.ibm.com","operator":"DoesNotExist"}]}]}}'
  serviceDefinition: data-cataloging-service-definition
  triggerUpdate: false
  doInstall: true
  enabled: true
EOF
```

Important: Ensure that you replace the `STORAGE_CLASS` details with the correct value.

b. Run the following command to apply the YAML.

```
oc apply -f deploy_datacataloging.yaml
```

c. Run the following command to verify whether the IBM Data Cataloging service is installed.

```
oc get FusionServiceInstance data-cataloging-service-instance -o json -n ibm-spectrum-fusion-ns
```

Backup & Restore

Backup & Restore server:

a. Create the YAML to deploy the Backup & Restore server.

```
cat << EOF > deploy_bnrserver.yaml
apiVersion: service.isf.ibm.com/v1
kind: FusionServiceInstance
metadata:
  name: ibm-backup-restore-service-instance
  namespace: ibm-spectrum-fusion-ns
spec:
  parameters:
    - name: namespace
      provided: false
      value: ibm-backup-restore
    - name: storageClass
      provided: true
      value: $STORAGE_CLASS$
    - name: doInstall
      provided: false
```

```
      value: 'true'
  serviceDefinition: ibm-backup-restore-service
  triggerUpdate: false
  doInstall: true
  enabled: true
EOF
```

Important:

- Ensure that you replace the `STORAGE_CLASS` details with the correct value.
- The Backup & Restore namespace must be less than 23 characters or shorter.
- Ensure that the Backup & Restore namespace must not be same namespace where base IBM Fusion is installed.

b. Run the following command to apply the YAML.

```
oc apply -f deploy_bnrserver.yaml
```

c. Run the following command to verify whether the Backup & Restore server is installed.

```
oc get FusionServiceInstance ibm-backup-restore-service-instance -o json -n ibm-spectrum-fusion-ns
```

Backup & Restore agent:

a. Create the YAML to deploy the Backup & Restore agent.

```
cat << EOF > deploy_bnragent.yaml
apiVersion: v1
kind: Secret
metadata:
  name: init-655759e819
  namespace: ibm-spectrum-fusion-ns
stringData:
  apiserver: $apiserver$
  bootstrapToken: $bootstrapToken$
---
apiVersion: application.isf.ibm.com/v1
kind: Connection
metadata:
  name: connection-655759e819
  namespace: ibm-spectrum-fusion-ns
spec:
  remoteCluster:
    apiEndpoint: $apiserver$
    initSecretName: init-655759e819
    connectionOperatorNamespace: ibm-spectrum-fusion-ns
---
apiVersion: service.isf.ibm.com/v1
kind: FusionServiceInstance
metadata:
  name: ibm-backup-restore-agent-service-instance
  namespace: ibm-spectrum-fusion-ns
spec:
  parameters:
  - name: namespace
    provided: false
    value: ibm-backup-restore
  - name: storageClass
    provided: true
    value: $STORAGE_CLASS$
  - name: hubEndPointURL
    provided: true
    value: $apiserver$
  - name: hubClusterName
    provided: true
    value: $hubClusterName$
  - name: connectionName
    provided: true
    value: connection-655759e819
  - name: doInstall
    provided: true
    value: "true"
  serviceDefinition: ibm-backup-restore-agent-service
  triggerUpdate: false
  doInstall: true
EOF
```

Important:

- The Backup & Restore namespace must be less than 23 characters or shorter.
- Ensure that the Backup & Restore namespace must not be same namespace where base IBM Fusion is installed.
- Ensure that you create a secret and connection with instance CR to install the Backup & Restore service.
- Ensure that the Backup & Restore server is installed on another cluster and generate the connection-token from that cluster.
- Ensure that you replace the `apiserver`, `bootstrapToken`, `STORAGE_CLASS`, and `hubClusterName` details with the correct values.

b. Run the following command to apply the YAML.

```
oc apply -f deploy_bnragent.yaml
```

c. Run the following command to verify whether the Backup & Restore agent is installed.

```
oc get FusionServiceInstance ibm-backup-restore-agent-service-instance -o json -n ibm-spectrum-fusion-ns
```

Disaster recovery

Disaster recovery offerings of IBM Fusion.

Disaster recovery (DR) is the ability to recover and continue business-critical applications from natural or human created disasters. It is a component of the overall business continuance strategy of any major organization as designed to preserve the continuity of business operations during major adverse events.

IBM Fusion with Fusion Data Foundation storage offers a resilient disaster recovery solution that ensures business continuity during data center unavailability with zero data loss and protects against regional failures, maintaining predictable data loss and business continuity. For more information, see Disaster recovery configuration for Fusion Data Foundation.

- **Disaster recovery configuration for Fusion Data Foundation**
  Disaster recovery offerings of IBM Fusion with Fusion Data Foundation storage.

Disaster recovery configuration for Fusion Data Foundation

Disaster recovery offerings of IBM Fusion with Fusion Data Foundation storage.

Disaster recovery provides the ability to fail over an application from one Red Hat® OpenShift® Container Platform cluster to another. It also allows relocating the same application back to the original primary cluster.

IBM Fusion with Fusion Data Foundation storage supports the following disaster recovery capabilities:

- **Metro-DR**
  Metropolitan disaster recovery (Metro-DR) ensures business continuity during the unavailability of a data center with no data loss. In the public cloud, it is similar to protecting from an Availability Zone failure. For more information, see Metro-DR configuration for Fusion Data Foundation.
- **Regional-DR**
  Regional disaster recovery (Regional-DR) ensures business continuity during the unavailability of a geographical region, accepting some loss of data in a predictable amount. In the public cloud, it is similar to protecting from a region failure. For more information, see Regional-DR configuration for Fusion Data Foundation.
- **Disaster recovery with stretch cluster**
  Stretch cluster solution ensures business continuity with no data loss disaster recovery protection by using Fusion Data Foundation-based synchronous replication. It operates within a single Red Hat OpenShift Container Platform cluster stretched across two low-latency data centers and one arbiter node. For more information, see Disaster recovery with stretch cluster for Fusion Data Foundation.

Zone failure in Metro-DR and region failure in Regional-DR in Regional-DR is expressed by using the terms, **Recovery Point Objective (RPO)** and **Recovery Time Objective (RTO)**.

- **RPO** is a measure of how frequently you take backups or snapshots of persistent data. In practice, the RPO indicates the amount of data that is lost or needs to be reentered after an outage.
- **RTO** is the amount of downtime a business can tolerate. The RTO answers the question, “How long can it take for our system to recover after we are notified of a business disruption?”

Deploying your disaster recovery solution

To protect your applications for disaster recovery (DR), deploy your DR solution as follows in the order mentioned here.

1. Deploy and configure IBM Fusion Data Foundation 4.16 Metro-DR or Regional-DR solution. For details, see Metro-DR configuration for Fusion Data Foundation or Regional-DR configuration for Fusion Data Foundation
2. Deploy IBM Fusion 2.8.x, as detailed in Deploying IBM Fusion

The IBM Metro- & Regional- Disaster Recovery instructions helps you to configure your infrastructure to failover an application from one OpenShift Container Platform cluster to another and then failback the same application to the original primary cluster. After securing your application, deploy Fusion as per the instructions given in step 2.

- **Metro-DR configuration for Fusion Data Foundation**
  This section of the guide provides you with insights into metropolitan disaster recovery (Metro-DR) configuration for Fusion Data Foundation to failover an application from one Red Hat OpenShift Container Platform cluster to another and then failback the same application to the original primary cluster.
- **Regional-DR configuration for Fusion Data Foundation**
  This section of the guide provides you with insights into regional disaster recovery (Regional-DR) configuration for Fusion Data Foundation to failover an application from one Red Hat OpenShift Container Platform cluster to another and then failback the same application to the original primary cluster.
- **Disaster recovery with stretch cluster for Fusion Data Foundation**
  This section of the guide provides you with insights into the IBM Fusion Data Foundation Disaster Recovery (DR) solution along with necessary configuration and recovery steps for stretch clusters.
- **Monitoring disaster recovery health**
  This section provides all the necessary configuration and commands for setting up the disaster recovery dashboard that help to monitor the health of your disaster recovery solution.
- **Troubleshooting disaster recovery**
  This troubleshooting section provides guidance or workarounds on how to fix some of the disaster recovery configuration issues.
- **Disabling disaster recovery for a disaster recovery-enabled application**
  Learn how to disable disaster recovery for disaster recovery-enabled managed and discovered applications.
- **Uninstalling disaster recovery from hub and managed clusters**
  Learn how to uninstall disaster recovery (DR) partially or completely from hub and managed clusters.

Metro-DR configuration for Fusion Data Foundation

This section of the guide provides you with insights into metropolitan disaster recovery (Metro-DR) configuration for Fusion Data Foundation to failover an application from one Red Hat® OpenShift® Container Platform cluster to another and then failback the same application to the original primary cluster.

In this case, the OpenShift Container Platform clusters are created or imported by using Red Hat Advanced Cluster Management (RHACM) with distance limitations between the OpenShift Container Platform clusters of less than 10 ms RTT latency. The persistent storage for applications is provided by an external IBM Storage Ceph cluster that is stretched between the two locations with the OpenShift Container Platform instances that are connected to this storage cluster. An arbiter node with a storage monitor service is required at a third location, which is a different location than where OpenShift Container Platform instances are deployed to establish quorum for the IBM Storage Ceph cluster in the case of a site outage. This third location can be in the range of ~100 ms RTT from the storage cluster that is connected to the OpenShift Container Platform instances.

This is a general overview of the Metro-DR steps that are required to configure and execute OpenShift application disaster recovery capabilities by using Fusion Data Foundation and RHACM across two distinct OpenShift Container Platform clusters separated by distance. In addition to these two managed clusters, a third OpenShift Container Platform cluster is required that is the RHACM hub cluster.

Important:
You can now set up Metropolitan disaster recovery solutions for workloads based on OpenShift virtualization technology using Fusion Data Foundation. For more information, see the knowledgebase article.

- **Components of Metro-DR solution**
  Metro-DR is composed of Red Hat Advanced Cluster Management for Kubernetes, IBM Storage Ceph, and Fusion Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters.
- **Unsupported features for Metro-DR**
  List of unsupported features in Metro-DR configuration.
- **Metro-DR deployment workflow**
  This section provides an overview of the steps required to configure and deploy Metro-DR capabilities using the latest versions of IBM Fusion Data Foundation, IBM Storage Ceph and Red Hat Advanced Cluster Management for Kubernetes (RHACM) version 2.9 or later, across two distinct OpenShift Container Platform clusters. In addition to two managed clusters, a third OpenShift Container Platform cluster is required to deploy the Advanced Cluster Management.
- **Requirements for enabling Metro-DR**
  This section lists all the prerequisites for implementing the Disaster Recovery solution supported by IBM Fusion Data Foundation.
- **Requirements for deploying IBM Storage Ceph stretch cluster with arbiter**
  IBM Storage Ceph is an open source enterprise platform that provides unified software-defined storage on standard, economical servers and disks. With block, object, and file storage that is combined into one platform, IBM Storage Ceph efficiently and automatically manages all your data, so you can focus on the applications and workloads that use it.
- **Deploying IBM Storage Ceph**
  Use this information for configuring your IBM Storage Ceph deployment.
- **Installing Fusion Data Foundation on managed clusters**
  To configure storage between the two OpenShift Container Platform clusters, Fusion Data Foundation operator must be installed first on each managed cluster.
- **Installing Fusion Data Foundation Multicluster Orchestrator operator**
  Fusion Data Foundation Multicluster Orchestrator is a controller that is installed from the Software Catalog of Red Hat OpenShift Container Platform on the hub cluster.
- **Configuring SSL access across clusters**
  Configure network (SSL) access between the Primary and Secondary clusters so that metadata can be stored on the alternate cluster in a Multicloud Gateway (MCG) object bucket using a secure transport protocol and in the Hub cluster for verifying access to the object buckets.
- **Creating Disaster Recovery Policy on Hub cluster**
  The Disaster Recovery Policy (DRPolicy) resource specifies OpenShift Container Platform clusters participating in the disaster recovery solution. DRPolicy is a cluster scoped resource that users can apply to applications that require Disaster Recovery solution.
- **Configure DRClusters for fencing automation**
  This configuration is required for enabling fencing prior to application failover. In order to prevent writes to the persistent volume from the cluster which is hit by a disaster, OpenShift DR instructs IBM Storage Ceph to fence the nodes of the cluster from the IBM Storage Ceph external storage. This section guides you on how to add the IPs or the IP Ranges for the nodes of the DRCluster.
- **Create sample application for testing disaster recovery solution**
  IBM Fusion Data Foundation disaster recovery (DR) solution supports disaster recovery for applications that are managed by Red Hat Advanced Cluster Management for Kubernetes (RHACM).
- **Subscription-based application failover between managed clusters**
  Use this application-based failover method when a managed cluster becomes unavailable, due to any reason.
- **ApplicationSet-based application failover between managed clusters**
  Use this application-based failover method when a managed cluster becomes unavailable, due to any reason.
- **Relocating Subscription-based application between managed clusters**
  Relocate an application to its preferred location when all managed clusters are available.
- **Relocating an ApplicationSet-based application between managed clusters**
  Relocate an application to its preferred location when all managed clusters are available.
- **Recovering to a replacement cluster with Metro-DR**
  When there is a failure with the primary cluster, you get the options to either repair, wait for the recovery of the existing cluster, or replace the cluster entirely if the cluster is irredeemable. This solution guides you when replacing a failed primary cluster with a new cluster and enables failback (relocate) to this new cluster.
- **Hub recovery using Red Hat Advanced Cluster Management [Technology preview]**
  Hub recovery using Red Hat Advanced Cluster Management enables you to use a passive hub cluster to failover or relocate disaster recovery protected workloads when the active hub is unavailable.

Components of Metro-DR solution

Metro-DR is composed of Red Hat Advanced Cluster Management for Kubernetes, IBM Storage Ceph, and Fusion Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters.

Red Hat Advanced Cluster Management for Kubernetes

Red Hat Advanced Cluster Management (RHACM) provides the ability to manage multiple clusters and application lifecycles. Hence, it serves as a control plane in a multi-cluster environment.

RHACM is split into two parts:

RHACM Hub

Components that run on the multi-cluster control plane.

Managed clusters
: Components that run on the clusters that are managed.

For more information about RHACM, see about this product, see About > Welcome to Red Hat Advanced Cluster Management for Kubernetes and Applications > Managing applications within Red Hat Advanced Cluster Management for Kubernetes product documentation.

IBM Storage Ceph

IBM Storage Ceph is a massively scalable, open, software-defined storage platform that combines the most stable version of the Ceph storage system with a Ceph management platform, deployment utilities, and support services. It significantly lowers the cost of storing enterprise data and helps organizations manage exponential data growth. The software is a robust and modern petabyte-scale storage platform for public or private cloud deployments.

For more information, see IBM Storage Ceph documentation.

Fusion Data Foundation

Fusion Data Foundation provides the ability to provision and manage storage for stateful applications in an OpenShift Container Platform cluster. It is backed by Ceph as the storage provider, whose lifecycle is managed by Rook in the Fusion Data Foundation component stack and Ceph-CSI provides the provisioning and management of Persistent Volumes for stateful applications.

OpenShift DR

OpenShift DR is a disaster recovery orchestrator for stateful applications across a set of peer OpenShift clusters, which are deployed and managed by using RHACM and provides cloud-native interfaces to orchestrate the life cycle of an application's state on Persistent Volumes. These include:

- Protecting an application and its state relationship across OpenShift clusters.
- Failing over an application and its state to a peer cluster.
- Relocate an application and its state to the previously deployed cluster.

OpenShift DR is split into three components:

IBM Fusion Data Foundation Multicluster Orchestrator
: Installed on the Hub cluster with RHACM, it orchestrates configuration and peering of Fusion Data Foundation clusters for Metro-DR relationships.

IBM Fusion Data Foundation DR Hub Operator
: Automatically installed as part of IBM Fusion Data Foundation Multicluster Orchestrator installation on the hub cluster to orchestrate failover or relocation of DR enabled applications.

IBM Fusion Data Foundation DR Cluster Operator
: Automatically installed on each managed cluster that is part of a Metro-DR relationship to manage the lifecycle of all PVCs of an application.

Unsupported features for Metro-DR

List of unsupported features in Metro-DR configuration.

The following features related to external mode are not supported in Metro-DR configuration:

- StorageClasses using non default RADOS namespace
- User created StorageClasses, even when using default RADOS namespace
- Multiple StorageClasses

Metro-DR deployment workflow

This section provides an overview of the steps required to configure and deploy Metro-DR capabilities using the latest versions of IBM Fusion Data Foundation, IBM Storage Ceph and Red Hat Advanced Cluster Management for Kubernetes (RHACM) version 2.9 or later, across two distinct OpenShift Container Platform clusters. In addition to two managed clusters, a third OpenShift Container Platform cluster is required to deploy the Advanced Cluster Management.

Procedure

To configure your infrastructure, perform the following steps:

1. Ensure requirements across the Hub, Primary and Secondary Openshift Container Platform clusters that are part of the DR solution are met.
   See Requirements for enabling Metro-DR.
2. Ensure you meet the requirements for deploying IBM Storage Ceph stretch cluster with arbiter.
   See Requirements for deploying IBM Storage Ceph stretch cluster with arbiter.
3. Deploy and configure IBM Storage Ceph stretch mode.
   For instructions on enabling Ceph cluster on two different data centers using stretched mode functionality, see Deploying IBM Storage Ceph.
4. Install Fusion Data Foundation operator and create a storage system on Primary and Secondary-managed clusters.
   See Installing Fusion Data Foundation on managed clusters.
5. Install the ODF Multicluster Orchestrator on the Hub cluster.
   See Installing Fusion Data Foundation Multicluster Orchestrator operator.
6. Configure SSL access between the Hub, Primary and Secondary clusters.
   See Configuring SSL access across clusters.
7. Create a DRPolicy resource for use with applications requiring DR protection across the Primary and Secondary clusters.
   See Creating Disaster Recovery Policy on Hub cluster.

Note: The Metro-DR solution can only have one DRpolicy.

8. Testing your disaster recovery solution with:
    a. Subscription-based application:
        - Create sample applications. See Creating sample application.
        - Test failover and relocate operations using the sample application between managed clusters. See Subscription-based application failover and relocating subscription-based application.

    b. ApplicationSet-based application:
        - Create sample applications. See Creating ApplicationSet-based applications.
        - Test failover and relocate operations using the sample application between managed clusters. See ApplicationSet-based application failover and relocating ApplicationSet-based application.

    c. Discovered applications
        - Ensure all requirements mentioned in Prerequisites is addressed. See Prerequisites for disaster recovery protection of discovered applications.
        - Create a sample discovered application. See Creating a sample discovered application
        - Enroll the discovered application. See Enrolling a sample discovered application for disaster recovery protection
        - Test failover and relocate. See Failover disaster recovery protected discovered application and Relocate disaster recovery protected discovered application.

Requirements for enabling Metro-DR

This section lists all the prerequisites for implementing the Disaster Recovery solution supported by IBM Fusion Data Foundation.

- You must have the following Red Hat OpenShift clusters that have network reachability between them:
    - **Hub cluster** where Red Hat Advanced Cluster Management (RHACM) for Kubernetes operator is installed.
    - **Primary-managed cluster** where Fusion Data Foundation is installed.
    - **Secondary-managed cluster** where Fusion Data Foundation is installed.

  Note: For configuring hub recovery setup, you need a 4th cluster which acts as the passive hub. The primary managed cluster (Site-1) can be co-situated with the active RHACM hub cluster while the passive hub cluster is situated along with the secondary managed cluster (Site-2). Alternatively, the active RHACM hub cluster can be placed in a neutral site (Site-3) that is not impacted by the failures of either of the primary managed cluster at Site-1 or the secondary cluster at Site-2. In this situation, if a passive hub cluster is used it can be placed with the secondary cluster at Site-2. For more information, see Configuring passive hub cluster.
  Hub recovery is a Technology Preview feature and is subject to Technology Preview support limitations.

- Ensure that RHACM operator and Multicluster Hub is installed on the Hub cluster. For detailed instructions, see the Install guide within Red Hat Advanced Cluster Management for Kubernetes product documentation.
  After the operator is successfully installed, a popover with a message that the Web console update is available appears on the user interface. Click **Refresh web console** from this popover for the console changes to reflect.
  Important: Ensure that application traffic routing and redirection are configured appropriately.
- On the Hub cluster,
    - Go to All Clusters > Infrastructure > Clusters.
    - Import or create the **Primary-managed cluster** and the **Secondary-managed cluster** by using the RHACM console.
    - Choose the appropriate options for your environment.

  For instructions, see Creating a cluster and Importing a target managed cluster to the hub cluster within the Clusters Red Hat Advanced Cluster Management for Kubernetes product documentation.

Warning: There are distance limitations between the locations where the OpenShift Container Platform managed clusters reside as well as the IBM Storage Ceph nodes. The network latency between the sites must be below 10-milliseconds round-trip time (RTT).

Requirements for deploying IBM Storage Ceph stretch cluster with arbiter

IBM Storage Ceph is an open source enterprise platform that provides unified software-defined storage on standard, economical servers and disks. With block, object, and file storage that is combined into one platform, IBM Storage Ceph efficiently and automatically manages all your data, so you can focus on the applications and workloads that use it.

This section provides a basic overview of the IBM Storage Ceph deployment. For more full deployment instructions, see IBM Storage Ceph documentation **6.1**.
Note: Only Flash media is supported since it runs with `min_size=1` when degraded. Use stretch mode only with all-flash OSDs. Using all-flash OSDs minimizes the time that is needed to recover once connectivity is restored, thus minimizing the potential for data loss.
Important: Erasure coded pools cannot be used with stretch mode.

- **Hardware requirements**
  Use this information for IBM Storage Ceph hardware requirements.
- **Software requirements**
  Use the latest software version of IBM Storage Ceph 6.1.
- **Network configuration requirements**
  Use this information for IBM Storage Ceph network configuration requirements.

Hardware requirements

Use this information for IBM Storage Ceph hardware requirements.

For information on minimum hardware requirements for deploying IBM Storage Ceph, see Planning > Hardware > Minimum hardware recommendations for containerized Ceph within IBM Storage Ceph documentation.

Table 1. Physical server locations and Ceph component layout for IBM Storage Ceph cluster

deployment

Node name	Datacenter	Ceph components
ceph1	DC1	OSD+MON+MGR
ceph2	DC1	OSD+MON
ceph3	DC1	OSD+MDS+RGW
ceph4	DC2	OSD+MON+MGR
ceph5	DC2	OSD+MON
ceph6	DC2	OSD+MDS+RGW
ceph7	DC3	MON

Software requirements

Use the latest software version of IBM Storage Ceph 6.1.

For more information on the supported operating system versions for IBM Storage Ceph, see **Compatibility matrix** within IBM Storage Ceph documentation.

Network configuration requirements

Use this information for IBM Storage Ceph network configuration requirements.

The recommended IBM Storage Ceph configuration is as follows:

- You must have two separate networks, one public network and one private network.
- You must have three different datacenters that support VLANS and subnets for Cephs private and public network for all datacenters.
  Note: You can use different subnets for each of the datacenters.
- The latencies between the two datacenters running the IBM Storage Ceph Object Storage Devices (OSDs) cannot exceed 10 ms RTT. For the **arbiter** datacenter, this was tested with values as high up to 100 ms RTT to the other two OSD datacenters.

Here is an example of a basic network configuration that are used in the documentation:

- **DC1: Ceph public/private network:** 10.0.40.0/24
- **DC2: Ceph public/private network:** 10.0.40.0/24
- **DC3: Ceph public/private network:** 10.0.40.0/24

For more information on the required IBM Storage Ceph network environment, see Configuring > Ceph network configuration within the IBM Storage Ceph documentation.

Deploying IBM Storage Ceph

Use this information for configuring your IBM Storage Ceph deployment.

- **Node pre-deployment steps**
  Before installing the IBM Storage Ceph cluster, be sure to fulfill all the requirements needed.
- **Cluster bootstrapping and service deployment with Cephadm**
  The cephadm utility installs and starts a single Ceph Monitor daemon and a Ceph Manager daemon for a new IBM Storage Ceph cluster on the local node where the cephadm bootstrap command is run. Use this information to bootstrap the cluster and deploy all the needed IBM Storage Ceph services in one step using a cluster specification yaml file.
- **Configuring IBM Storage Ceph stretch mode**
  After the IBM Storage Ceph cluster is fully deployed using `cephadm`, use this information to configure the stretch cluster mode. The new stretch mode is designed to handle the 2-site case.

Node pre-deployment steps

Before installing the IBM Storage Ceph cluster, be sure to fulfill all the requirements needed.

Procedure

Perform the following steps to fullfill all requirements.

1. Register all the nodes to the Red Hat Network or Red Hat Satellite and subscribe to a valid pool:

   ```
   subscription-manager register
   subscription-manager subscribe --pool=8a8XXXXXX9e0
   ```

2. Enable access for all the nodes in the Ceph cluster for the following repositories:
   - `rhel-9-for-x86_64-baseos-rpms`
   - `rhel-9-for-x86_64-appstream-rpms`

   ```
   subscription-manager repos --disable="*" --enable="rhel-9-for-x86_64-baseos-rpms" --enable="rhel-9-for-x86_64-appstream-rpms"
   ```

3. Update the operating system RPMs to the latest version and reboot, if needed.

```
dnf update -y
```

```
reboot
```

4. Select a node from the cluster to be your bootstrap node.
ceph1 is the bootstrap node in this example. Only on the bootstrap node **ceph1**, enable the **ansible-2.9-for-rhel-9-x86_64-rpms** and **rhceph-5-tools-for-rhel-9-x86_64-rpms** repositories:

```
subscription-manager repos --enable="ansible-2.9-for-rhel-9-x86_64-rpms" --enable="rhceph-5-tools-for-rhel-9-x86_64-rpms"
```

5. Configure the **hostname** using the bare/short hostname in all the hosts.

```
hostnamectl set-hostname <short_name>
```

6. Verify the hostname configuration for deploying IBM Storage Ceph with cephadm.

```
hostname
```

Example output: `ceph1`

7. Modify /etc/hosts file and add the fqdn entry to the 127.0.0.1 IP by setting the *DOMAIN* variable with our DNS domain name.

```
DOMAIN="example.domain.com"

cat <<EOF >/etc/hosts
127.0.0.1 $(hostname).${DOMAIN} $(hostname) localhost localhost.localdomain localhost4 localhost4.localdomain4
::1       $(hostname).${DOMAIN} $(hostname) localhost6 localhost6.localdomain6
EOF
```

8. Check the long hostname with the fqdn using the **hostname -f** option.

```
hostname -f
```

Example output:

```
ceph1.example.domain.com
```

Note: To understand more about these required changes, see Fully qualified domain names vs bare host names within Ceph product documentation.

9. Run the following steps on bootstrap node.
ceph1 is the bootstrap node in this example.
   a. Install the cephadm-ansible RPM package:

   ```
   sudo dnf install -y cephadm-ansible
   ```

   Important: To run the Ansible playbooks, you must have **ssh** passwordless access to all the nodes that are configured to the IBM Storage Ceph cluster. Ensure that the configured user (for example, **deployment-user**) has root privileges to invoke the **sudo** command without needing a password.
   b. Use a custom key.
   Configure the selected user (for example, **deployment-user**) ssh config file to specify the id or key that is used for connecting to the nodes via ssh.

   ```
   cat <<EOF > ~/.ssh/config
   Host ceph*
      User deployment-user
      IdentityFile ~/.ssh/ceph.pem
   EOF
   ```

   c. Build the Ansible inventory.

   ```
   cat <<EOF > /usr/share/cephadm-ansible/inventory
   ceph1
   ceph2
   ceph3
   ceph4
   ceph5
   ceph6
   ceph7
   [admin]
   ceph1
   ceph4
   EOF
   ```

   Note: Here, the Hosts (**Ceph1** and **Ceph4**) belonging to two different data centers are configured as part of the [admin] group on the inventory file and are tagged as **_admin** by **cephadm**. Each of these admin nodes receive the admin ceph keyring during the bootstrap process so that when one data center is down, we can check using the other available admin node.
   d. Verify that **ansible** can access all nodes using ping module before running the pre-flight playbook.

   ```
   ansible -i /usr/share/cephadm-ansible/inventory -m ping all -b
   ```

   Example output:

   ```
   ceph6 | SUCCESS => {
       "ansible_facts": {
           "discovered_interpreter_python": "/usr/libexec/platform-python"
       },
       "changed": false,
       "ping": "pong"
   }
   ceph4 | SUCCESS => {
       "ansible_facts": {
           "discovered_interpreter_python": "/usr/libexec/platform-python"
       },
       "changed": false,
       "ping": "pong"
   ```

```
}
ceph3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ceph2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ceph5 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ceph1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ceph7 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
```

e. Go to the /usr/share/cephadm-ansible directory.
f. Run **ansible-playbook** with relative file paths.

```
ansible-playbook -i /usr/share/cephadm-ansible/inventory /usr/share/cephadm-ansible/cephadm-preflight.yml --extra-vars "ceph_origin=rhcs"
```

The preflight playbook Ansible playbook configures the IBM Storage Ceph `dnf` repository and prepares the storage cluster for bootstrapping. It also installs podman, lvm2, chronyd, and cephadm. The default location for cephadm-ansible and cephadm-preflight.yml is /usr/share/cephadm-ansible.
For more information, see Installing > IBM Storage Ceph installation > Running the preflight playbook within IBM Storage Ceph documentation.

Cluster bootstrapping and service deployment with Cephadm

The cephadm utility installs and starts a single Ceph Monitor daemon and a Ceph Manager daemon for a new IBM Storage Ceph cluster on the local node where the cephadm bootstrap command is run. Use this information to bootstrap the cluster and deploy all the needed IBM Storage Ceph services in one step using a cluster specification yaml file.

About this task

If you find issues during the deployment, troubleshoot the errors by dividing the deployment into two steps:

- Bootstrap
- Service deployment

Note: For more information about the bootstrapping process, see Installing > IBM Storage Ceph installation > Bootstrapping a new storage cluster within IBM Storage Ceph documentation.

Procedure

1. Create a JSON file to authenticate against the container registry.
   For example:

```
cat <<EOF > /root/registry.json
{
 "url":"registry.redhat.io",
 "username":"User",
 "password":"Pass"
}
EOF
```

2. Create a cluster-spec.yaml that adds the nodes to the IBM Storage Ceph cluster and also sets specific labels for where the services should run.
   Note: Be sure to follow specifications as detailed in Table 1.

```
cat <<EOF > /root/cluster-spec.yaml
service_type: host
addr: 10.0.40.78  ## <XXX.XXX.XXX.XXX>
hostname: ceph1   ##  <ceph-hostname-1>
location:
  root: default
  datacenter: DC1
```

```
labels:
  - osd
  - mon
  - mgr
---
service_type: host
addr: 10.0.40.35
hostname: ceph2
location:
  datacenter: DC1
labels:
  - osd
  - mon
---
service_type: host
addr: 10.0.40.24
hostname: ceph3
location:
  datacenter: DC1
labels:
  - osd
  - mds
  - rgw
---
service_type: host
addr: 10.0.40.185
hostname: ceph4
location:
  root: default
  datacenter: DC2
labels:
  - osd
  - mon
  - mgr
---
service_type: host
addr: 10.0.40.88
hostname: ceph5
location:
  datacenter: DC2
labels:
  - osd
  - mon
---
service_type: host
addr: 10.0.40.66
hostname: ceph6
location:
  datacenter: DC2
labels:
  - osd
  - mds
  - rgw
---
service_type: host
addr: 10.0.40.221
hostname: ceph7
labels:
  - mon
---
service_type: mon
placement:
  label: "mon"
---
service_type: mds
service_id: cephfs
placement:
  label: "mds"
---
service_type: mgr
service_name: mgr
placement:
  label: "mgr"
---
service_type: osd
service_id: all-available-devices
service_name: osd.all-available-devices
placement:
  label: "osd"
spec:
  data_devices:
    all: true
---
service_type: rgw
service_id: objectgw
service_name: rgw.objectgw
placement:
  count: 2
  label: "rgw"
spec:
  rgw_frontend_port: 8080
EOF
```

3. Retrieve the IP for the NIC with the IBM Storage Ceph public network configured from the bootstrap node.
   After substituting `10.0.40.0` with the subnet that you have defined in your ceph public network, run the following command:

```
ip a | grep 10.0.40
```

Example output: `10.0.40.78`

4. Run the **Cephadm** bootstrap command as the **root** user on the node that is the initial Monitor node in the cluster.
   The *IP_ADDRESS* option is the node's IP address that you are using to run the **cephadm bootstrap** command.
   Note: If you have configured a different user instead of `root` for passwordless SSH access, then use the--ssh-user= flag with the **cepadm bootstrap** command.
   If you are using non default or id_rsa ssh key names, then use --ssh-private-key and --ssh-public-key options with **cephadm** command.

```
cephadm  bootstrap --ssh-user=deployment-user --mon-ip 10.0.40.78 --apply-spec /root/cluster-spec.yaml --registry-json
/root/registry.json
```

   Important: If the local node uses fully-qualified domain names (FQDN), then add the --allow-fqdn-hostname option to cephadm bootstrap on the command line.
   After the bootstrap finishes, the following output appears from the previous **cephadm bootstrap** command:

```
You can access the Ceph CLI with:

        sudo /usr/sbin/cephadm shell --fsid dd77f050-9afe-11ec-a56c-029f8148ea14 -c /etc/ceph/ceph.conf -k
/etc/ceph/ceph.client.admin.keyring

Please consider enabling telemetry to help improve Ceph:

        ceph telemetry on

For more information see:

        https://docs.ceph.com/docs/pacific/mgr/telemetry/
```

5. Verify the status of IBM Storage Ceph cluster deployment using the Ceph CLI client from ceph1.

```
ceph -s
```

   Example output:

```
cluster:
  id:     3a801754-e01f-11ec-b7ab-005056838602
  health: HEALTH_OK

services:
  mon: 5 daemons, quorum ceph1,ceph2,ceph4,ceph5,ceph7 (age 4m)
  mgr: ceph1.khuuot(active, since 5m), standbys: ceph4.zotfsp
  osd: 12 osds: 12 up (since 3m), 12 in (since 4m)
  rgw: 2 daemons active (2 hosts, 1 zones)

data:
  pools:   5 pools, 107 pgs
  objects: 191 objects, 5.3 KiB
  usage:   105 MiB used, 600 GiB / 600 GiB avail
           105 active+clean
```

   Note: It may take several minutes for all the services to start. It is normal to get a global recovery event while you don't have any OSDs configured. You can use **ceph orch ps** and **ceph orch ls** commands to further check the status of the services.
6. Verify if all the nodes are part of the `cephadm` cluster.

```
ceph orch host ls
```

   Example output:

```
HOST   ADDR          LABELS  STATUS
ceph1  10.0.40.78    _admin osd mon mgr
ceph2  10.0.40.35    osd mon
ceph3  10.0.40.24    osd mds rgw
ceph4  10.0.40.185   osd mon mgr
ceph5  10.0.40.88    osd mon
ceph6  10.0.40.66    osd mds rgw
ceph7  10.0.40.221   mon
```

   Note: You can run Ceph commands directly from the host because `ceph1` was configured in the `cephadm-ansible` inventory as part of the [admin] group. The Ceph admin keys were copied to the host during the **cephadm bootstrap** process.
7. Check the current placement of the Ceph monitor services on the datacenters.

```
ceph orch ps | grep mon | awk '{print $1 " " $2}'
```

   Example output:

```
mon.ceph1 ceph1
mon.ceph2 ceph2
mon.ceph4 ceph4
mon.ceph5 ceph5
mon.ceph7 ceph7
```

8. Check the current placement of the Ceph manager services on the datacenters.

```
ceph orch ps | grep mgr | awk '{print $1 " " $2}'
```

   Example output:

```
mgr.ceph2.ycgwyz ceph2
mgr.ceph5.kremtt ceph5
```

9. Check the ceph osd crush map layout to ensure that each host has one OSD configured and its status is *UP*.
   Note: Be sure to check that each node is under the correct datacenter bucket, as detailed in Table 1.

```
ceph osd tree
```

Example output:

```
ID   CLASS  WEIGHT   TYPE NAME           STATUS  REWEIGHT  PRI-AFF
-1          0.87900  root default
-16         0.43950      datacenter DC1
-11         0.14650          host ceph1
  2    ssd  0.14650              osd.2       up   1.00000  1.00000
 -3         0.14650          host ceph2
  3    ssd  0.14650              osd.3       up   1.00000  1.00000
-13         0.14650          host ceph3
  4    ssd  0.14650              osd.4       up   1.00000  1.00000
-17         0.43950      datacenter DC2
 -5         0.14650          host ceph4
  0    ssd  0.14650              osd.0       up   1.00000  1.00000
 -9         0.14650          host ceph5
  1    ssd  0.14650              osd.1       up   1.00000  1.00000
 -7         0.14650          host ceph6
  5    ssd  0.14650              osd.5       up   1.00000  1.00000
```

10. Create and enable a new RDB block pool.

```
ceph osd pool create rbdpool 32 32
```

```
ceph osd pool application enable rbdpool rbd
```

Note: The number 32 at the end of the command is the number of PGs assigned to this pool. The number of PGs can vary depending on several factors like the number of OSDs in the cluster, expected % used of the pool, etc. You can use the following calculator to determine the number of PGs needed: Ceph Placement Groups (PGs) per Pool Calculator.

11. Verify that the RBD pool has been created.

```
ceph osd lspools | grep rbdpool
```

Example output: `3 rbdpool`

12. Verify that MDS services are active and has located one service on each datacenter.

```
ceph orch ps | grep mds
```

Example output:

```
mds.cephfs.ceph3.cjpbqo    ceph3               running (17m)   117s ago  17m    16.1M        -  16.2.9
mds.cephfs.ceph6.lqmgqt    ceph6               running (17m)   117s ago  17m    16.1M        -  16.2.9
```

13. Create the CephFS volume.

```
ceph fs volume create cephfs
```

Note: The `ceph fs volume create` command also creates the needed data and meta CephFS pools.

14. Check the `Ceph` status to verify how the MDS daemons have been deployed.
Ensure that the state is active where `ceph6` is the primary MDS for this filesystem and `ceph3` is the secondary MDS.

```
ceph fs status
```

Example output:

```
cephfs - 0 clients
======
RANK  STATE           MDS              ACTIVITY     DNS    INOS   DIRS   CAPS
 0    active  cephfs.ceph6.ggjywj  Reqs:    0 /s    10     13     12      0
       POOL           TYPE     USED  AVAIL
cephfs.cephfs.meta  metadata  96.0k   284G
cephfs.cephfs.data    data       0    284G
    STANDBY MDS
cephfs.ceph3.ogcqkl
```

15. Verify that RGW services are active.

```
ceph orch ps | grep rgw
```

Example output:

```
rgw.objectgw.ceph3.kkmxgb  ceph3  *:8080       running (7m)      3m ago   7m    52.7M        -  16.2.9
rgw.objectgw.ceph6.xmnpah  ceph6  *:8080       running (7m)      3m ago   7m    53.3M        -  16.2.9
```

Configuring IBM Storage Ceph stretch mode

After the IBM Storage Ceph cluster is fully deployed using `cephadm`, use this information to configure the stretch cluster mode. The new stretch mode is designed to handle the 2-site case.

Procedure

1. Check the current election strategy being used by the monitors with the **ceph mon dump** command.
By default in a ceph cluster, the connectivity is set to classic.

```
ceph mon dump | grep election_strategy
```

Example output:

```
dumped monmap epoch 9
election_strategy: 1
```

2. Change the monitor election to connectivity.

```
ceph mon set election_strategy connectivity
```

3. Rerun the **ceph mon dump** command to verify the `election_strategy` value.

```
ceph mon dump | grep election_strategy
```

Example output:

```
dumped monmap epoch 10
election_strategy: 3
```

For more information about the different election strategies, see Administering > Operations > Management of monitors > Configuring monitor election strategy within the IBM Storage Ceph documentation.

4. Set the location for all Ceph monitors.

```
ceph mon set_location ceph1 datacenter=DC1
ceph mon set_location ceph2 datacenter=DC1
ceph mon set_location ceph4 datacenter=DC2
ceph mon set_location ceph5 datacenter=DC2
ceph mon set_location ceph7 datacenter=DC3
```

5. Verify that each monitor has its appropriate location.

```
ceph mon dump
```

Example output:

```
epoch 17
fsid dd77f050-9afe-11ec-a56c-029f8148ea14
last_changed 2022-03-04T07:17:26.913330+0000
created 2022-03-03T14:33:22.957190+0000
min_mon_release 16 (pacific)
election_strategy: 3
0: [v2:10.0.143.78:3300/0,v1:10.0.143.78:6789/0] mon.ceph1; crush_location {datacenter=DC1}
1: [v2:10.0.155.185:3300/0,v1:10.0.155.185:6789/0] mon.ceph4; crush_location {datacenter=DC2}
2: [v2:10.0.139.88:3300/0,v1:10.0.139.88:6789/0] mon.ceph5; crush_location {datacenter=DC2}
3: [v2:10.0.150.221:3300/0,v1:10.0.150.221:6789/0] mon.ceph7; crush_location {datacenter=DC3}
4: [v2:10.0.155.35:3300/0,v1:10.0.155.35:6789/0] mon.ceph2; crush_location {datacenter=DC1}
```

6. Create a CRUSH rule that makes use of this OSD crush topology by installing the `ceph-base` RPM package in order to use the **crushtool** command.

```
dnf -y install ceph-base
```

For more information about CRUSH ruleset, see Concepts > Architecture > Core Ceph components > Ceph CRUS ruleset within the IBM Storage Ceph documentation.

7. Get the compiled CRUSH map from the cluster.

```
ceph osd getcrushmap > /etc/ceph/crushmap.bin
```

8. Decompile the CRUSH map and convert it to a text file in order to be able to edit it.

```
crushtool -d /etc/ceph/crushmap.bin -o /etc/ceph/crushmap.txt
```

9. Add the following rule to the CRUSH map by editing the text file /etc/ceph/crushmap.txt at the end of the file.

```
vim /etc/ceph/crushmap.txt
```

```
rule stretch_rule {
        id 1
        type replicated
        min_size 1
        max_size 10
        step take default
        step choose firstn 0 type datacenter
        step chooseleaf firstn 2 type host
        step emit
}
# end crush map
```

This example is applicable for active applications in both OpenShift Container Platform clusters.
Note: The rule `id` has to be unique. In this example, there is only one more CRUSH rule with `id 0` hence we are using `id 1`. If your deployment has more rules created, then use the next free ID.
The CRUSH rule declared contains the following information:

`Rule name`
: A unique whole name for identifying the rule. In this example, `stretch_rule`.

`id`
: A unique whole number for identifying the rule. In this example, `1`.

`type`
: A rule for either a storage drive replicated or erasure-coded. In this example, `replicated`.

`min_size`
: If a pool makes fewer replicas than this number, CRUSH does not select this rule. In this example, `1`.

`max_size`
: If a pool makes more replicas than this number, CRUSH does not select this rule. In this example, `10`.

`step take default`
: Takes the root bucket called `default`, and begins iterating down the tree.

`step choose firstn 0 type datacenter`
Selects the datacenter bucket, and goes into it's subtrees.
`step chooseleaf firstn 2 type host`
Selects the number of buckets of the given type. In this case, it is two different hosts located in the datacenter it entered at the previous level.
`step emit`
Outputs the current value and empties the stack. Typically used at the end of a rule, but may also be used to pick from different trees in the same rule.

10. Compile the new CRUSH map from the file /etc/ceph/crushmap.txt and convert it to a binary file called /etc/ceph/crushmap2.bin.

```
crushtool -c /etc/ceph/crushmap.txt -o /etc/ceph/crushmap2.bin
```

11. Inject the newly created CRUSH map back into the cluster.

```
ceph osd setcrushmap -i /etc/ceph/crushmap2.bin
```

Example output: `17`
Note: The number 17 is a counter and it increases (18,19, and so on) depending on the changes made to the CRUSH map.
12. Verify that the stretched rule created is now available for use.

```
ceph osd crush rule ls
```

Example output:

```
replicated_rule
stretch_rule
```

13. Enable the stretch cluster mode.

```
ceph mon enable_stretch_mode ceph7 stretch_rule datacenter
```

In this example, `ceph7` is the arbiter node, `stretch_rule` is the crush rule we created in the previous step and `datacenter` is the dividing bucket.
14. Verify all our pools are using the `stretch_rule` CRUSH rule that was created as part of the Ceph cluster.

```
for pool in $(rados lspools);do echo -n "Pool: ${pool}; ";ceph osd pool get ${pool} crush_rule;done
```

Example output:

```
Pool: device_health_metrics; crush_rule: stretch_rule
Pool: cephfs.cephfs.meta; crush_rule: stretch_rule
Pool: cephfs.cephfs.data; crush_rule: stretch_rule
Pool: .rgw.root; crush_rule: stretch_rule
Pool: default.rgw.log; crush_rule: stretch_rule
Pool: default.rgw.control; crush_rule: stretch_rule
Pool: default.rgw.meta; crush_rule: stretch_rule
Pool: rbdpool; crush_rule: stretch_rule
```

This indicates that a working IBM Storage Ceph stretched cluster with arbiter mode is now available.

Installing Fusion Data Foundation on managed clusters

To configure storage between the two OpenShift Container Platform clusters, Fusion Data Foundation operator must be installed first on each managed cluster.

Before you begin

Ensure that you have met the hardware requirements for Fusion Data Foundation external deployments. For a detailed description of the hardware requirements, see External mode requirements.

Procedure

1. Install and configure the latest **Fusion Data Foundation** cluster on each of the managed clusters.
2. After installing the operator, create a StorageSystem using the Full deployment type and Connect with external storage platform, where your Backing storage type is IBM Storage Ceph.
For more information, see Deploying Data Foundation in external mode.

   a. Use the following flags with the ceph-external-cluster-details-exporter.py script.
   - At a minimum, you must use the following three flags.

     `--rbd-data-pool-name`
     With the name of the RBD pool that was created during IBM Storage Ceph deployment for OpenShift Container Platform. For example, the pool can be called `rbdpool`.
     `--rgw-endpoint`
     Provide the endpoint in the format `<ip_address>:<port>`. It is the RGW IP of the RGW daemon running on the same site as the OpenShift Container Platform cluster that you are configuring.
     `--run-as-user`
     With a different client name for each site.

   - The following flags are `optional` if default values were used during the IBM Storage Ceph deployment.

     `--cephfs-filesystem-name`
     With the name of the CephFS file system created during IBM Storage Ceph deployment for OpenShift Container Platform, the default file system name is `cephfs`.
     `--cephfs-data-pool-name`
     With the name of the CephFS datapool created during IBM Storage Ceph deployment for OpenShift Container Platform, the default pool is called `cephfs.data`.

`--cephfs-metadata-pool-name`
With the name of the CephFS metadata pool created during IBM Storage Ceph deployment for OpenShift Container Platform, the default pool is called `cephfs.meta`.

b. Run the following command on the bootstrap node, ceph1, to get the IP for the RGW endpoints in datacenter1 and datacenter2.

```
ceph orch ps | grep rgw.objectgw
```

Example output:

```
rgw.objectgw.ceph3.mecpzm  ceph3  *:8080       running (5d)     31s ago   7w     204M        -  16.2.7-112.el8cp
rgw.objectgw.ceph6.mecpzm  ceph6  *:8080       running (5d)     31s ago   7w     204M        -  16.2.7-112.el8cp
```

```
host ceph3.example.com
host ceph6.example.com
```

Example output:

```
ceph3.example.com has address 10.0.40.24
ceph6.example.com has address 10.0.40.66
```

c. Run the ceph-external-cluster-details-exporter.py with the parameters that are configured for the first OpenShift Container Platform managed cluster `cluster1` on bootstrapped node `ceph1` .

```
python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name rbdpool --cephfs-filesystem-name cephfs --
cephfs-data-pool-name cephfs.cephfs.data  --cephfs-metadata-pool-name cephfs.cephfs.meta --<rgw-endpoint>
XXX.XXX.XXX.XXX:8080 --run-as-user client.odf.cluster1 > ocp-cluster1.json
```

Note:
Modify the `<rgw-endpoint> XXX.XXX.XXX.XXX` according to your environment.

d. Run the ceph-external-cluster-details-exporter.py with the parameters that are configured for the first OpenShift Container Platform managed cluster `cluster2` on bootstrapped node `ceph1`.

```
python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name rbdpool --cephfs-filesystem-name cephfs --
cephfs-data-pool-name cephfs.cephfs.data  --cephfs-metadata-pool-name cephfs.cephfs.meta --rgw-endpoint
XXX.XXX.XXX.XXX:8080 --run-as-user client.odf.cluster2 > ocp-cluster2.json
```

Note:
Modify the `<rgw-endpoint> XXX.XXX.XXX.XXX` according to your environment.

e. Save the two files generated in the bootstrap cluster (ceph1) ocp-cluster1.json and ocp-cluster2.json to your local machine.
f. Use the contents of file ocp-cluster1.json on the OpenShift Container Platform console on cluster1 where external Fusion Data Foundation is being deployed.
g. Use the contents of file ocp-cluster2.json on the OpenShift Container Platform console on cluster2 where external Fusion Data Foundation is being deployed.

3. Review the settings and then select Create StorageSystem.
4. Validate the successful deployment of Fusion Data Foundation on each managed cluster with the following command:

```
oc get storagecluster -n openshift-storage ocs-external-storagecluster -o jsonpath='{.status.phase}{"\n"}'
```

For the Multicloud Gateway (MCG):

```
oc get noobaa -n openshift-storage noobaa -o jsonpath='{.status.phase}{"\n"}'
```

Wait for the status result to be *Ready* for both queries on the **Primary-managed cluster** and the **Secondary-managed cluster**.

What to do next

From the Red Hat OpenShift Web Console, navigate to Installed Operators > Fusion Data Foundation > Storage System > ocs-storagecluster-storagesystem > Resources and verify that the Status of `StorageCluster` is *Ready* and has a green tick mark next to it.

Installing Fusion Data Foundation Multicluster Orchestrator operator

Fusion Data Foundation Multicluster Orchestrator is a controller that is installed from the Software Catalog of Red Hat® OpenShift® Container Platform on the hub cluster.

Before you begin

- Ensure that you created the following `CatalogSource` manifest on the hub cluster:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: isf-data-foundation-catalog
  namespace: openshift-marketplace
spec:
  displayName: Data Foundation Catalog
  grpcPodConfig:
    affinity:
      nodeAffinity:
        preferredDuringSchedulingIgnoredDuringExecution:
          - preference:
              matchExpressions:
                - key: gpu.isf.ibm.com
                  operator: DoesNotExist
            weight: 50
    securityContextConfig: restricted
  icon:
```

```
    base64data: ""
    mediatype: ""
  image: icr.io/cpopen/isf-data-foundation-catalog:v4.21
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 60m
```

Note: Ensure that the version number you specify in the `image` field of the `CatalogSource` manifest matches the Fusion Data Foundation version installed on the managed cluster.

- Ensure that you are logged in to the Red Hat OpenShift Container Platform as an administrator.
- Ensure that `ibm-spectrum-fusion-ns` is selected in the Project field.

Procedure

1. Go to Ecosystem > Software Catalog and use the keyword filter to search for `Multicluster Orchestrator`.
2. Click the IBM-provided Multicluster Orchestrator tile.
3. Keep all default settings and click Install.
   Ensure that the operator resources are installed in `openshift-operators` project and available to all namespaces.
   Note: The `IBM Fusion Data Foundation Multicluster Orchestrator` also installs the **DR Hub Operator** on the RHACM hub cluster as a dependency.
4. Verify that the operator **Pods** are in a *Running* state.
   The DR Hub operator is also installed at the same time in `openshift-operators` namespace.

   ```
   oc get pods -n openshift-operators
   ```

   Example output:

   ```
   NAME                                          READY   UP-TO-DATE   AVAILABLE   AGE
   odf-multicluster-console-6845b795b9-blxrn     1/1     Running      0           4d20h
   odfmo-controller-manager-f9d9dfb59-jbrsd      1/1     Running      0           4d20h
   ramen-hub-operator-6fb887f885-fss4w           2/2     Running      0           4d20h
   ```

Configuring SSL access across clusters

Configure network (SSL) access between the Primary and Secondary clusters so that metadata can be stored on the alternate cluster in a Multicloud Gateway (MCG) object bucket using a secure transport protocol and in the Hub cluster for verifying access to the object buckets.

About this task

If all of your OpenShift clusters are deployed using a signed and valid set of certificates for your environment then this section can be skipped. Proceed to Creating Disaster Recovery Policy on Hub cluster.

Procedure

1. Extract the ingress certificate for the Primary-managed cluster and save the output to primary.crt.

   ```
   oc get cm default-ingress-cert -n openshift-config-managed -o jsonpath="{['data']['ca-bundle\.crt']}" > primary.crt
   ```

2. Extract the ingress certificate for the Secondary-managed cluster and save the output to secondary.crt.

   ```
   oc get cm default-ingress-cert -n openshift-config-managed -o jsonpath="{['data']['ca-bundle\.crt']}" > secondary.crt
   ```

3. Create a new ConfigMap file to hold the remote cluster's certificate bundle with filename cm-clusters-crt.yaml.
   Note: There could be more or less than three certificates for each cluster, as shown in this example file. Also, ensure that the certificate contents are correctly indented after you copy and paste from the primary.crt and secondary.crt files that were created before.

   ```
   apiVersion: v1
   data:
     ca-bundle.crt: |
       -----BEGIN CERTIFICATE-----
       <copy contents of cert1 from primary.crt here>
       -----END CERTIFICATE-----

       -----BEGIN CERTIFICATE-----
       <copy contents of cert2 from primary.crt here>
       -----END CERTIFICATE-----

       -----BEGIN CERTIFICATE-----
       <copy contents of cert3 primary.crt here>
       -----END CERTIFICATE----

       -----BEGIN CERTIFICATE-----
       <copy contents of cert1 from secondary.crt here>
       -----END CERTIFICATE-----

       -----BEGIN CERTIFICATE-----
       <copy contents of cert2 from secondary.crt here>
       -----END CERTIFICATE-----

       -----BEGIN CERTIFICATE-----
       <copy contents of cert3 from secondary.crt here>
   ```

```
    -----END CERTIFICATE-----
kind: ConfigMap
metadata:
  name: user-ca-bundle
  namespace: openshift-config
```

4. Create the ConfigMap on the Primary-managed cluster, Secondary-managed cluster, and the Hub cluster.

   ```
   oc create -f cm-clusters-crt.yaml
   ```

   Example output:

   ```
   configmap/user-ca-bundle created
   ```

5. Patch default proxy resource on the Primary-managed cluster, Secondary-managed cluster, and the Hub cluster.

   ```
   oc patch proxy cluster --type=merge  --patch='{"spec":{"trustedCA":{"name":"user-ca-bundle"}}}'
   ```

   Example output:

   ```
   proxy.config.openshift.io/cluster patched
   ```

Creating Disaster Recovery Policy on Hub cluster

The Disaster Recovery Policy (DRPolicy) resource specifies OpenShift Container Platform clusters participating in the disaster recovery solution. DRPolicy is a cluster scoped resource that users can apply to applications that require Disaster Recovery solution.

Before you begin

Ensure that there is a minimum set of two managed clusters.

About this task

The Fusion Data Foundation MultiCluster Orchestrator Operator facilitates the creation of each DRPolicy and the corresponding DRClusters through the Multicluster Web console.

Procedure

1. On the **OpenShift console**, navigate to All Clusters > Data Services > Disaster recovery.
2. On the Overview tab, click Create a disaster recovery policy or you can navigate to Policies tab and click Create DRPolicy.
3. Enter Policy name.
   Ensure that each DRPolicy has a unique name (for example: `ocp4perf1-ocp4perf2`).
4. Select two clusters from the list of managed clusters with which this new policy is associated.
5. Replication policy is automatically set to `sync`, based on the OpenShift clusters selected.
6. Click Create.
7. Verify that the DRPolicy is created successfully.
   Run this command on the Hub cluster for each of the DRPolicy resources created, where *<drpolicy_name>* is replaced with your unique name.

   ```
   oc get drpolicy <drpolicy_name> -o jsonpath='{.status.conditions[].reason}{"\n"}'
   ```

   Example output: `Succeeded`

   When a DRPolicy is created, along with it, two DRCluster resources are also created. It might take up to 10 minutes for all three resources to be validated and for the status to show as `Succeeded`.

   Note: Editing of `SchedulingInterval`, `ReplicationClassSelector`, `VolumeSnapshotClassSelector` and `DRClusters` field values are not supported in the DRPolicy.
8. Verify the object bucket access from the Hub cluster to both the **Primary-managed cluster** and the **Secondary-managed cluster**.
   a. Get the names of the DRClusters on the Hub cluster.

      ```
      oc get drclusters
      ```

      Example output:

      ```
      NAME         AGE
      ocp4perf1   4m42s
      ocp4perf2   4m42s
      ```

   b. Check S3 access to each bucket created on **each** managed cluster.
      Use the DRCluster validation command, where *<drcluster_name>* is replaced with your unique name.
      Note: Editing of `Region` and `S3ProfileName` field values are non supported in DRClusters.

      ```
      oc get drcluster <drcluster_name> -o jsonpath='{.status.conditions[2].reason}{"\n"}'
      ```

      Example output: `Succeeded`

      Note: Ensure to run command for both DRClusters on the Hub cluster.
9. Verify that the IBM Fusion Data Foundation DR Cluster operator installation was successful on the **Primary-managed cluster** and the **Secondary-managed cluster**.

   ```
   oc get csv,pod -n openshift-dr-system
   ```

   Example output:

```
NAME                                                                  DISPLAY                          VERSION
REPLACES   PHASE
clusterserviceversion.operators.coreos.com/odr-cluster-operator.v4.14.0          Openshift DR Cluster Operator    4.14.0
Succeeded
clusterserviceversion.operators.coreos.com/volsync-product.v0.8.0                VolSync                          0.8.0
Succeeded
```

```
NAME                                           READY   STATUS    RESTARTS   AGE
pod/ramen-dr-cluster-operator-6467cf5d4c-cc8kz   2/2     Running   0          3d12h
```

You can also verify that IBM Fusion Data Foundation DR Cluster Operator is installed successfully on the **OperatorHub** of each managed cluster.

10. Verify that the secret is propagated correctly on the Primary managed cluster and the Secondary managed cluster.

    ```
    oc get secrets -n openshift-dr-system | grep Opaque
    ```

    Match the output with the s3SecretRef from the Hub cluster:

    ```
    oc get cm -n openshift-operators ramen-hub-operator-config -oyaml
    ```

Configure DRClusters for fencing automation

This configuration is required for enabling fencing prior to application failover. In order to prevent writes to the persistent volume from the cluster which is hit by a disaster, OpenShift DR instructs IBM Storage Ceph to fence the nodes of the cluster from the IBM Storage Ceph external storage. This section guides you on how to add the IPs or the IP Ranges for the nodes of the DRCluster.

- **Add node IP addresses to DRClusters**
  Add node IP addresses to DRClusters, enabling fencing prior to application failover.
- **Add fencing annotations to DRClusters**
  Add fencing annotations to DRClusters, enabling fencing prior to application failover.

Add node IP addresses to DRClusters

Add node IP addresses to DRClusters, enabling fencing prior to application failover.

Procedure

1. Find the IP addresses for all of the OpenShift nodes in the managed clusters by running this command in the Primary-managed cluster and the Secondary-managed cluster.

   ```
   oc get nodes -o jsonpath='{range .items[*]}{.status.addresses[?(@.type=="ExternalIP")].address}{"\n"}{end}'
   ```

   Example output:

   ```
   10.70.56.118
   10.70.56.193
   10.70.56.154
   10.70.56.242
   10.70.56.136
   10.70.56.99
   ```

   After you have the IP addresses then the **DRCluster** resources can be modified for each managed cluster.

2. Find the DRCluster names on the Hub Cluster.

   ```
   oc get drcluster
   ```

   Example output:

   ```
   NAME        AGE
   ocp4perf1   5m35s
   ocp4perf2   5m35s
   ```

3. Edit each DRCluster to add your unique IP addresses, replacing *<drcluster_name>* with your unique name.

   ```
   oc edit drcluster <drcluster_name>
   ```

   ```
   apiVersion: ramendr.openshift.io/v1alpha1
   kind: DRCluster
   metadata:
   [...]
   spec:
     s3ProfileName: s3profile-<drcluster_name>-ocs-external-storagecluster
     ## Add this section
     cidrs:
       -  <IP_Address1>/32
       -  <IP_Address2>/32
       -  <IP_Address3>/32
       -  <IP_Address4>/32
       -  <IP_Address5>/32
       -  <IP_Address6>/32
   [...]
   ```

   Example output:

```
drcluster.ramendr.openshift.io/ocp4perf1 edited
```

Note: There could be more than six IP addresses.
Modify this DRCluster configuration also for IP addresses on the Secondary-managed clusters in the peer DRCluster resource (for example, ocp4perf2).

Add fencing annotations to DRClusters

Add fencing annotations to DRClusters, enabling fencing prior to application failover.

Procedure

1. Add the following annotations to all the DRCluster resources.
   These annotations include details the NetworkFence resource uses before testing application failover.

   Use this **edit** command, replacing *<drcluster_name>* with your unique name.

   ```
   oc edit drcluster <drcluster_name>
   ```

   ```
   apiVersion: ramendr.openshift.io/v1alpha1
   kind: DRCluster
   metadata:
     ## Add this section
     annotations:
       drcluster.ramendr.openshift.io/storage-clusterid: openshift-storage
       drcluster.ramendr.openshift.io/storage-driver: openshift-storage.rbd.csi.ceph.com
       drcluster.ramendr.openshift.io/storage-secret-name: rook-csi-rbd-provisioner
       drcluster.ramendr.openshift.io/storage-secret-namespace: openshift-storage
   [...]
   ```

   Example output:

   ```
   drcluster.ramendr.openshift.io/ocp4perf1 edited
   ```

2. Ensure to add these annotations for both DRCluster resources (for example: `ocp4perf1` and `ocp4perf2`).

Create sample application for testing disaster recovery solution

IBM Fusion Data Foundation disaster recovery (DR) solution supports disaster recovery for applications that are managed by Red Hat® Advanced Cluster Management for Kubernetes (RHACM).

The following section detail how to create an application and apply a DRPolicy to an application:

- ApplicationSet-based applications
  OpenShift users that do not have cluster-admin permissions cannot create ApplicationSet-based applications. For more information on ApplicationSet-based application, see ApplicationSet in *Red Hat Advanced Cluster Management for Kubernetes product documentation*.

Important: Discovered applications using RBD block volumes and Ceph FileSystem are not supported when using Metro-DR.

- **Deleting a sample application**
  The sample application `busybox` can be deleted by using the Red Hat Advanced Cluster Management for Kubernetes (RHACM) console.

Creating ApplicationSet-based applications

Create an ApplicationSet-based application on the Hub cluster using the Argo CD Pull model to test Metro-DR failover and relocate across managed clusters.

Before you begin

- Ensure that the Red Hat OpenShift GitOps operator is installed on all three clusters: **Hub cluster**, **Primary managed cluster** and **Secondary managed cluster**. For instructions, see Installing Red Hat OpenShift GitOps Operator in web console.
- On the Hub cluster, ensure that both Primary and Secondary managed clusters are registered to GitOps. For registration instructions, see Application management and GitOps > GitOps > Registering managed clusters to Red Hat OpenShift GitOps operator in Red Hat Advanced Cluster Management for Kubernetes product documentation. Then check if the Placement used by `GitOpsCluster` resource to register both managed clusters, has the tolerations to deal with cluster unavailability. You can verify if the following tolerations are added to the Placement using the command.

  ```
  oc get placement <placement-name> -n openshift-gitops -o yaml.
  ```

  ```
    tolerations:
    - key: cluster.open-cluster-management.io/unreachable
      operator: Exists
    - key: cluster.open-cluster-management.io/unavailable
      operator: Exists
  ```

  In case the tolerations are not added, see Application management and GitOps > GitOps in Red Hat Advanced Cluster Management for Kubernetes product documentation.
- Ensure that you have created the `ClusterRoleBinding` yaml on both the **Primary** and **Secondary managed** clusters. For instruction, see Application management and GitOps > GitOps > Deploying Argo CD with Push and Pull model > Prerequisites in Red Hat Advanced Cluster Management for Kubernetes product documentation.

Procedure

1. On the Hub cluster, navigate to All Clusters > Applications > Create application.
2. Select type as Argo CD ApplicationSet - Pull model.
3. In General step, enter your Application set name.
4. Select Argo server `openshift-gitops` and Requeue time as `180` seconds.
5. Click Next.
6. In the Repository location for resources section, select Repository type Git.
7. Enter the Git repository URL for the sample application, the GitHub Branch, and Path where the resources `busybox` Pod and PVC are created.
    a. Use the sample application repository as https://github.com/red-hat-storage/ocm-ramen-samples
    b. Select Revision as release-4.16 or higher
    c. Choose Path as busybox-odr-metro
8. Enter Remote namespace value. (example, busybox-sample) and click Next.
9. Select Sync policy settings and click Next.
   You can choose one or more options.
10. Add a label *<name>* with its value set to the managed cluster name.
11. Click Next.
12. Review the setting details and click Submit.

Apply Data policy to sample ApplicationSet-based application

Apply a data policy to an ApplicationSet-based application on the Hub cluster to enable disaster recovery protection and verify that `ClusterDataProtected` is set to true.

Before you begin

Ensure that both managed clusters referenced in the DRPolicy are reachable. If not, the application is not DR protected until both clusters are online.

Procedure

1. On the Hub cluster, navigate to All Clusters > Applications.
2. Click the Actions menu at the end of DRPolicy that you want to use to view the list of available actions.
3. Click Manage data policy > Assign Data Policy.
4. Select Policy and click Next.
5. Select an Application resource and then use PVC label selector to select `PCV label` for the selected resource.
   Note: You can select more than one PVC label for the selected application resources.
6. After adding all the application resources, click Next.
7. Review the **Policy configuration details** and click Assign. The newly assigned Data policy is displayed on the **Manage data policy** modal list view.
8. Verify that you can view the assigned policy details on the Applications page.
    a. On the Applications page, navigate to the Data policy column and click the policy link to expand the view.
    b. Verify that you can see the number of policies assigned along with failover and relocate status.
9. After you apply DRPolicy to the applications, confirm whether the `ClusterDataProtected` is set to `True` in the drpc yaml output.

Deleting a sample application

The sample application `busybox` can be deleted by using the Red Hat Advanced Cluster Management for Kubernetes (RHACM) console.

Before you begin

Important: When deleting a DR protected application, access to both clusters that belong to the DRPolicy is required. This is to ensure that all protected API resources and resources in the respective S3 stores are cleaned up as part of removing the DR protection. If access to one of the clusters is not healthy, deleting the `DRPlacementControl` resource for the application, on the hub, would remain in the `Deleting` state.
Do not begin deleting a sample application until the failover and relocate testing is completed and the application is ready to be removed from RHACM and the managed clusters.

About this task

Procedure

1. On the **Hub cluster**, navigate to Applications.
2. Search for the sample application to be deleted (for example, `busybox`).
3. Click the Action menu next to the application that you want to delete.
4. Click Delete application.
   When the Delete application is selected a new screen appears asking if the application-related resources should also be deleted.
5. Select Remove application related resources to delete the Subscription and PlacementRule.
6. Click Delete.
   This deletes the `busybox` application on the Primary-managed cluster (or the cluster that the application was running on).
7. In addition to the resources deleted by using the RHACM console, delete the `DRPlacementControl` if it is not auto-deleted after deleting the `busybox` application.

a. Log in to the Red Hat OpenShift web console for the Hub cluster and navigate to Installed Operators for the project **busybox-sample**.
For ApplicationSet applications, select the project as **openshift-gitops**.
b. Click OpenShift DR Hub Operator > DRPlacementControl.
c. Click the Action menu next to the **busybox** application DRPlacementControl that you want to delete.
d. Click Delete DRPlacementControl.
e. Click Delete.

Note: This process can be used to delete any application with a **DRPlacementControl** resource.

Subscription-based application failover between managed clusters

Use this application-based failover method when a managed cluster becomes unavailable, due to any reason.

Before you begin

- If your setup has active and passive RHACM hub clusters, see Hub recovery using Red Hat Advanced Cluster Management [Technology preview].
- When primary cluster is in a state other than Ready, check the actual status of the cluster as it might take some time to update.
  1. Go to RHACM console > Infrastructure > Clusters > Cluster list tab.
  2. Check the status of both the managed clusters individually before performing a failover operation.

  However, failover operation can still be run when the cluster you are failing over to is in a *Ready* state.

Procedure

1. Enable fencing on the **Hub cluster**.
   a. Open CLI terminal and edit the DRCluster resource, where *<drcluster_name>* is your unique name.
   CAUTION:
   After the managed cluster is fenced, **all** communication from applications to the Fusion Data Foundation external storage cluster fails and some Pods are in an unhealthy state (for example, **CreateContainerError**, **CrashLoopBackOff**) on the cluster that is now fenced.

   ```
   oc edit drcluster <drcluster_name>

   apiVersion: ramendr.openshift.io/v1alpha1
   kind: DRCluster
   metadata:
   [...]
   spec:
     ## Add this line
     clusterFence: Fenced
     cidrs:
     [...]
   [...]
   ```

   Example output:

   ```
   drcluster.ramendr.openshift.io/ocp4perf1 edited
   ```

   b. Verify the fencing status on the **Hub cluster** for the **Primary-managed cluster**, replacing *<drcluster_name>* is your unique identifier.

   ```
   oc get drcluster.ramendr.openshift.io <drcluster_name> -o jsonpath='{.status.phase}{"\n"}'
   ```

   Example output: `Fenced`
   c. Verify that the IPs that belong to the OpenShift Container Platform cluster nodes are now in the blocklist.

   ```
   ceph osd blocklist ls
   ```

   Example output

   ```
   cidr:10.1.161.1:0/32 2028-10-30T22:30:03.585634+0000
   cidr:10.1.161.14:0/32 2028-10-30T22:30:02.483561+0000
   cidr:10.1.161.51:0/32 2028-10-30T22:30:01.272267+0000
   cidr:10.1.161.63:0/32 2028-10-30T22:30:05.099655+0000
   cidr:10.1.161.129:0/32 2028-10-30T22:29:58.335390+0000
   cidr:10.1.161.130:0/32 2028-10-30T22:29:59.861518+0000
   ```

2. On the Hub cluster, navigate to Applications.
3. Click the Actions menu at the end of application row to view the list of available actions.
4. Click Failover application.
5. After the **Failover application** popup is shown, select Policy and Target cluster to which the associated application fails over in a disaster.
6. Click the Select subscription group dropdown to verify the default selection or modify this setting.
   By default, the subscription group that replicates for the application resources is selected.
7. Check the status of the Failover readiness.
   - If the status is *Ready* with a green tick, it indicates that the target cluster is ready for failover to start. Proceed to step 8.
   - If the status is *Unknown* or *Not ready*, then wait until the status changes to *Ready*.
8. Click Initiate.
   All the system workloads and their available resources are now transferred to the target cluster.
9. Close the modal window and track the status using the Data policy column on the Applications page.
10. Verify that the activity status shows as *FailedOver* for the application.
    a. Go to Applications > Overview.
    b. In the Data policy column, click the policy link for the application you applied the policy to.
    c. On the Data Policy popover page, click the View more details link.

ApplicationSet-based application failover between managed clusters

Use this application-based failover method when a managed cluster becomes unavailable, due to any reason.

Before you begin

- If your setup has active and passive RHACM hub clusters, see Hub recovery using Red Hat Advanced Cluster Management [Technology preview].
- When primary cluster is in a state other than Ready, check the actual status of the cluster as it might take some time to update.
  1. Go to RHACM console > Infrastructure > Clusters > Cluster list tab.
  2. Check the status of both the managed clusters individually before performing a failover operation.

  However, failover operation can still be run when the cluster you are failing over to is in a *Ready* state.

Procedure

1. Enable fencing on the Hub cluster.
   a. Open CLI terminal and edit the DRCluster resource, where *<drcluster_name>* is your unique name.
   CAUTION:
   After the managed cluster is fenced, **all** communication from applications to the Fusion Data Foundation external storage cluster fails and some Pods are in an unhealthy state (for example, `CreateContainerError`, `CrashLoopBackOff`) on the cluster that is now fenced.

   ```
   oc edit drcluster <drcluster_name>

   apiVersion: ramendr.openshift.io/v1alpha1
   kind: DRCluster
   metadata:
   [...]
   spec:
     ## Add this line
     clusterFence: Fenced
     cidrs:
     [...]
   [...]
   ```

   Example output:

   ```
   drcluster.ramendr.openshift.io/ocp4perf1 edited
   ```

   b. Verify the fencing status on the Hub cluster for the **Primary-managed cluster**, replacing *<drcluster_name>* is your unique identifier.

   ```
   oc get drcluster.ramendr.openshift.io <drcluster_name> -o jsonpath='{.status.phase}{"\n"}'
   ```

   Example output: `Fenced`
   c. Verify that the IPs that belong to the OpenShift Container Platform cluster nodes are now in the blocklist.

   ```
   ceph osd blocklist ls
   ```

   Example output

   ```
   cidr:10.1.161.1:0/32 2028-10-30T22:30:03.585634+0000
   cidr:10.1.161.14:0/32 2028-10-30T22:30:02.483561+0000
   cidr:10.1.161.51:0/32 2028-10-30T22:30:01.272267+0000
   cidr:10.1.161.63:0/32 2028-10-30T22:30:05.099655+0000
   cidr:10.1.161.129:0/32 2028-10-30T22:29:58.335390+0000
   cidr:10.1.161.130:0/32 2028-10-30T22:29:59.861518+0000
   ```

2. On the Hub cluster, navigate to Applications.
3. Click the Actions menu at the end of application row to view the list of available actions.
4. Click Failover application.
5. When the **Failover application** modal is shown, verify the details that are presented are correct and check the status of the Failover readiness. If the status is *Ready* with a green tick, it indicates that the target cluster is ready for failover to start.
6. Click Initiate.
   All the system workloads and their available resources are now transferred to the target cluster.
7. Close the modal window and track the status using the Data policy column on the Applications page.
8. Verify that the activity status shows as *FailedOver* for the application.
   a. Go to Applications > Overview.
   b. In the Data policy column, click the policy link for the application you applied the policy to.
   c. On the Data Policy popover page, verify that you can see one or more policy names and the relocation status that is associated with the policy in use with the application.

Relocating Subscription-based application between managed clusters

Relocate an application to its preferred location when all managed clusters are available.

Before you begin

- If your setup has active and passive RHACM hub clusters, see Hub recovery using Red Hat Advanced Cluster Management [Technology preview].
- When primary cluster is in a state other than Ready, check the actual status of the cluster as it might take some time to update. Relocate can only be performed when both primary and preferred clusters are up and running.
  1. Go to RHACM console > Infrastructure > Clusters > Cluster list tab.

2. Check the status of both the managed clusters individually before performing a relocate operation.

- Verify that applications were cleaned up from the cluster before unfencing it.

Procedure

1. Disable fencing on the Hub cluster.
   a. Edit the **DRCluster resource** for this cluster, replacing *<drcluster_name>* with your unique name.

   ```
   oc edit drcluster <drcluster_name>

   apiVersion: ramendr.openshift.io/v1alpha1
   kind: DRCluster
   metadata:
   [...]
   spec:
     cidrs:
     [...]
     ## Modify this line
     clusterFence: Unfenced
     [...]
   [...]
   ```

   Example output:

   ```
   drcluster.ramendr.openshift.io/ocp4perf1 edited
   ```

   b. Gracefully restart OpenShift Container Platform nodes that were `Fenced`.
   A restart is required to resume the I/O operations after unfencing to avoid any further recovery orchestration failures. Restart all nodes of the cluster by following the steps in the Nodes > Working with nodes > Understanding node rebooting > Rebooting a node gracefully Red Hat OpenShift Container Platform product documentation.
   Note: Ensure that all the nodes are initially cordoned and drained before you restart and perform uncordon operations on the nodes.
   c. After all Red Hat OpenShift nodes are rebooted and are in a *Ready* status, verify that all Pods are in a healthy state.
   Run the following command on the **Primary-managed cluster** (or the cluster has been Unfenced):

   ```
   oc get pods -A | egrep -v 'Running|Completed'
   ```

   Example output:

   ```
   NAMESPACE                                          NAME
   READY   STATUS      RESTARTS       AGE
   ```

   The output for this query is zero Pods before proceeding to the next step.
   Note: If there are Pods still in an unhealthy status because of severed storage communication, troubleshoot and resolve before continuing. Because the storage cluster is external to Red Hat OpenShift, it also must be properly recovered after a site outage for Red Hat OpenShift applications to be healthy. Alternatively, you can use the Red Hat OpenShift Web Console dashboards and Overview tab to assess the health of applications and the external Fusion Data Foundation storage cluster. The detailed Fusion Data Foundation dashboard is found by going to Storage > Data Foundation.

   d. Verify that the `Unfenced` cluster is in a healthy state.
   Validate the fencing status in the Hub cluster for the Primary-managed cluster, replacing *<drcluster_name>* with your unique name.

   ```
   oc get drcluster.ramendr.openshift.io <drcluster_name> -o jsonpath='{.status.phase}{"\n"}'
   ```

   Example output: `Unfenced`
   e. Verify that the IPs that belong to the OpenShift Container Platform cluster nodes are NOT in the blocklist.

   ```
   ceph osd blocklist ls
   ```

   Ensure that you do not see the IPs added during fencing.

2. On the Hub cluster, navigate to Applications.
3. Click the Actions menu at the end of application row to view the list of available actions.
4. Click Relocate application.
5. When the Relocate application modal is shown, select Policy and Target cluster to which the associated application migrates to.
6. By default, the subscription group that replicates the application resources is selected. Click the Select subscription group dropdown to verify the default selection or modify this setting.
7. Check the status of the Relocation readiness.
   - If the status is *Ready* with a green tick, it indicates that the target cluster is ready for failover to start. Proceed to step 8.
   - If the status is *Unknown* or *Not ready*, then wait until the status changes to *Ready*.
8. Click Initiate.
   All the system workloads and their available resources are now transferred to the target cluster.
9. Close the modal window and track the status using the Data policy column on the Applications page.
10. Verify that the activity status shows as *Relocated* for the application.
    a. Go to Applications > Overview.
    b. In the Data policy column, click the policy link for the application you applied the policy to.
    c. On the Data Policy popover page, click the View more details link.

Relocating an ApplicationSet-based application between managed clusters

Relocate an application to its preferred location when all managed clusters are available.

Before you begin

- If your setup has active and passive RHACM hub clusters, see Hub recovery using Red Hat Advanced Cluster Management [Technology preview].
- When primary cluster is in a state other than Ready, check the actual status of the cluster as it might take some time to update. Relocate can only be performed when both primary and preferred clusters are up and running.
    1. Go to RHACM console > Infrastructure > Clusters > Cluster list tab.
    2. Check the status of both the managed clusters individually before performing a relocate operation.
- Verify that applications were cleaned up from the cluster before unfencing it.

Procedure

1. Disable fencing on the Hub cluster.
    a. Edit the **DRCluster resource** for this cluster, replacing *<drcluster_name>* with a unique name.

    ```
    oc edit drcluster <drcluster_name>

    apiVersion: ramendr.openshift.io/v1alpha1
    kind: DRCluster
    metadata:
    [...]
    spec:
      cidrs:
      [...]
      ## Modify this line
      clusterFence: Unfenced
      [...]
    [...]
    ```

    Example output:

    ```
    drcluster.ramendr.openshift.io/ocp4perf1 edited
    ```

    b. Gracefully reboot OpenShift Container Platform nodes that were `Fenced`.
    A restart is required to resume the I/O operations after unfencing to avoid any further recovery orchestration failures. Restart all nodes of the cluster by following the steps in the Nodes > Working with nodes > Understanding node rebooting > Rebooting a node gracefully Red Hat OpenShift Container Platform product documentation.
    Note: Ensure that all the nodes are initially cordoned and drained before you restart and perform uncordon operations on the nodes.
    c. After all Red Hat OpenShift nodes are rebooted and are in a *Ready* status, verify that all Pods are in a healthy state.
    Run the following command on the **Primary-managed cluster** (or the cluster has been Unfenced):

    ```
    oc get pods -A | egrep -v 'Running|Completed'
    ```

    Example output:

    ```
    NAMESPACE                                          NAME
    READY   STATUS      RESTARTS       AGE
    ```

    The output for this query is zero Pods before proceeding to the next step.
    Note: If there are Pods still in an unhealthy status because of severed storage communication, troubleshoot and resolve before continuing. Because the storage cluster is external to Red Hat OpenShift, it also must be properly recovered after a site outage for Red Hat OpenShift applications to be healthy. Alternatively, you can use the Red Hat OpenShift Web Console dashboards and Overview tab to assess the health of applications and the external Fusion Data Foundation storage cluster. The detailed Fusion Data Foundation dashboard is found by going to Storage > Data Foundation.

    d. Verify that the `Unfenced` cluster is in a healthy state.
    Validate the fencing status in the Hub cluster for the Primary-managed cluster, replacing *<drcluster_name>* with your unique name.

    ```
    oc get drcluster.ramendr.openshift.io <drcluster_name> -o jsonpath='{.status.phase}{"\n"}'
    ```

    Example output: `Unfenced`
    e. Verify that the IPs that belong to the OpenShift Container Platform cluster nodes are NOT in the blocklist.

    ```
    ceph osd blocklist ls
    ```

    Ensure that you do not see the IPs added during fencing.

2. On the Hub cluster, navigate to Applications.
3. Click the Actions menu at the end of application row to view the list of available actions.
4. Click Relocate application.
5. When the Relocate application modal is shown, select Policy and Target cluster to which the associated application migrates to.
6. Click Initiate.
   All the system workloads and their available resources are now transferred to the target cluster.
7. Close the modal window and track the status using the Data policy column on the Applications page.
8. Verify that the activity status shows as *Relocated* for the application.
    a. Go to Applications > Overview.
    b. In the Data policy column, click the policy link for the application you applied the policy to.
    c. On the Data Policy popover page, verify that you can see one or more policy names and the relocation status associated with the policy in use with the application.

Recovering to a replacement cluster with Metro-DR

When there is a failure with the primary cluster, you get the options to either repair, wait for the recovery of the existing cluster, or replace the cluster entirely if the cluster is irredeemable. This solution guides you when replacing a failed primary cluster with a new cluster and enables failback (relocate) to this new cluster.

Before you begin

- Ensure that the Metro-DR environment has been configured with applications installed using Red Hat Advance Cluster Management (RHACM).
- Ensure that the applications are assigned a Data policy which protects them against cluster failure.

About this task

In these instructions, we are assuming that a RHACM managed cluster must be replaced after the applications have been installed and protected. For purposes of this section, the RHACM managed cluster is the **replacement cluster**, while the cluster that is not replaced is the **surviving cluster** and the new cluster is the **recovery cluster**.
Note: Replacement cluster recovery for Discovered applications is currently not supported. Only Managed applications are supported.

Procedure

1. Perform the following steps on the **Hub cluster**:
   a. Fence the replacement cluster by using the CLI terminal to edit the **DRCluster** resource, where *<drcluster_name>* is the replacement cluster name.

   ```
   oc edit drcluster <drcluster_name>

   apiVersion: ramendr.openshift.io/v1alpha1
   kind: DRCluster
   metadata:
   [...]
   spec:
     ## Add or modify this line
     clusterFence: Fenced
     cidrs:
     [...]
   [...]
   ```

   b. Using the RHACM console, navigate to Applications and failover all protected applications from the failed cluster to the surviving cluster.
   c. Verify and ensure that all protected applications are now running on the surviving cluster.
   Note: The **PROGRESSION** state for each application DRPlacementControl shows as `Cleaning Up`. This is expected if the replacement cluster is offline or down.
2. Unfence the replacement cluster.
   Using the CLI terminal, edit the DRCluster resource, where *<drcluster_name>* is the replacement cluster name.

   ```
   oc edit drcluster <drcluster_name>

   apiVersion: ramendr.openshift.io/v1alpha1
   kind: DRCluster
   metadata:
   [...]
   spec:
     ## Modify this line
     clusterFence: Unfenced
     cidrs:
     [...]
   [...]
   ```

3. Delete the DRCluster for the replacement cluster.

   ```
   oc delete drcluster <drcluster_name> --wait=false
   ```

   Note: Use `--wait=false` since the DRCluster is not deleted until a later step.
4. Disable disaster recovery on the **Hub cluster** for each protected application on the surviving cluster.
   a. For each application, edit the Placement and ensure that the surviving cluster is selected.
   Note: For Subscription-based applications the associated Placement can be found in the same namespace on the hub cluster similar to the managed clusters. For ApplicationSets-based applications the associated Placement can be found in the `openshift-gitops` namespace on the hub cluster.

   ```
   oc edit placement <placement_name> -n <namespace>

   apiVersion: cluster.open-cluster-management.io/v1beta1
   kind: Placement
   metadata:
   annotations:
     cluster.open-cluster-management.io/experimental-scheduling-disable: "true"
   [...]
   spec:
   clusterSets:
   - submariner
   predicates:
   - requiredClusterSelector:
       claimSelector: {}
       labelSelector:
         matchExpressions:
         - key: name
           operator: In
           values:
           - cluster1  <-- Modify to be surviving cluster name
   [...]
   ```

   b. Verify that the `s3Profile` is removed for the replacement cluster by running the following command on the surviving cluster for each protected application's VolumeReplicationGroup.

   ```
   oc get vrg -n <application_namespace> -o jsonpath='{.items[0].spec.s3Profiles}' | jq
   ```

   c. After the protected application Placement resources are all configured to use the surviving cluster and replacement cluster s3Profile(s) removed from protected applications, all DRPlacementControl resources must be deleted from the Hub cluster.

   ```
   oc delete drpc <drpc_name> -n <namespace>
   ```

Note: For Subscription-based applications the associated DRPlacementControl can be found in the same namespace as the managed clusters on the hub cluster. For ApplicationSets-based applications the associated DRPlacementControl can be found in the `openshift-gitops` namespace on the hub cluster.

d. Verify that all DRPlacementControl resources are deleted before proceeding to the next step. This command is a query across all namespaces. There should be no resources found.

```
oc get drpc -A
```

e. The last step is to edit each applications Placement and remove the annotation `cluster.open-cluster-management.io/experimental-scheduling-disable: "true"`.

```
oc edit placement <placement_name> -n <namespace>
```

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
annotations:
  ## Remove this annotation
  cluster.open-cluster-management.io/experimental-scheduling-disable: "true"
[...]
```

5. Repeat the process detailed in the last step and the sub-steps for every protected application on the surviving cluster. Disabling DR for protected applications is now completed.
6. On the Hub cluster, run the following script to remove all disaster recovery configurations from the **surviving cluster** and the **hub cluster**.

```
#!/bin/bash
secrets=$(oc get secrets -n openshift-operators | grep Opaque | cut -d" " -f1)
echo $secrets
for secret in $secrets
do
    oc patch -n openshift-operators secret/$secret -p '{"metadata":{"finalizers":null}}' --type=merge
done
mirrorpeers=$(oc get mirrorpeer -o name)
echo $mirrorpeers
for mp in $mirrorpeers
do
    oc patch $mp -p '{"metadata":{"finalizers":null}}' --type=merge
    oc delete $mp
done
drpolicies=$(oc get drpolicy -o name)
echo $drpolicies
for drp in $drpolicies
do
    oc patch $drp -p '{"metadata":{"finalizers":null}}' --type=merge
    oc delete $drp
done
drclusters=$(oc get drcluster -o name)
echo $drclusters
for drp in $drclusters
do
    oc patch $drp -p '{"metadata":{"finalizers":null}}' --type=merge
    oc delete $drp
done
oc delete project openshift-operators
managedclusters=$(oc get managedclusters -o name | cut -d"/" -f2)
echo $managedclusters
for mc in $managedclusters
do
    secrets=$(oc get secrets -n $mc | grep multicluster.odf.openshift.io/secret-type | cut -d" " -f1)
    echo $secrets
    for secret in $secrets
    do
        set -x
        oc patch -n $mc secret/$secret -p '{"metadata":{"finalizers":null}}' --type=merge
        oc delete -n $mc secret/$secret
    done
done

oc delete clusterrolebinding spoke-clusterrole-bindings
```

Note: This script used the command `oc delete project openshift-operators` to remove the Disaster Recovery (DR) operators in this namespace on the hub cluster. If there are other non-DR operators in this namespace, you must install them again from OperatorHub.

7. After the namespace openshift-operators is automatically created again, add the monitoring label back for collecting the disaster recovery metrics.

```
oc label namespace openshift-operators openshift.io/cluster-monitoring='true'
```

8. On the surviving cluster, ensure that the object bucket created during the DR installation is deleted. Delete the object bucket if it was not removed by script. The name of the object bucket used for DR starts with `odrbucket`.

```
oc get obc -n openshift-storage
```

9. On the RHACM console, navigate to Infrastructure > Clusters view.
   a. Detach the replacement cluster.
   b. Create a new OpenShift cluster (recovery cluster) and import the new cluster into the RHACM console. For instructions, see see Creating clusters and Cluster import within the Cluster management and multicluster engine > Clusters guide in Red Hat Advanced Cluster Management for Kubernetes product documentation.
10. Install Fusion Data Foundation operator on the recovery cluster and connect it to the same external IBM Storage Ceph as the surviving cluster. For detailed instructions, refer to Deploying Data Foundation in external mode.
    Note:
    Ensure that the Fusion Data Foundation version is 4.15 or later and the same version is on the surviving cluster.

11. On the hub cluster, install the ODF Multicluster Orchestrator operator from OperatorHub. For instructions, see chapter on Installing Fusion Data Foundation on managed clusters.
12. Using the RHACM console, navigate to Data Services > Data policies.
    a. Select Create DRPolicy and name your policy.
    b. Select the recovery cluster and the surviving cluster.
    c. Create the policy. For instructions see chapter on Creating Disaster Recovery Policy on Hub cluster.
    Proceed to the next step only after the status of DRPolicy changes to `Validated`.
13. Apply the DRPolicy to the applications on the surviving cluster that were originally protected before the replacement cluster failed.
14. Relocate the newly protected applications on the surviving cluster back to the new recovery (primary) cluster. Using the RHACM console, navigate to the Applications menu to perform the relocation.

Hub recovery using Red Hat Advanced Cluster Management [Technology preview]

Hub recovery using Red Hat Advanced Cluster Management enables you to use a passive hub cluster to failover or relocate disaster recovery protected workloads when the active hub is unavailable.

When your setup has active and passive Red Hat Advanced Cluster Management for Kubernetes (RHACM) hub clusters, and in case where the active hub is down, you can use the passive hub to failover or relocate the disaster recovery protected workloads.
Important:
Hub recovery is a Technology Preview feature and is subject to Technology Preview support limitations. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

- **Configuring passive hub cluster**
  Configure the passive hub cluster in a Metro DR environment so that disaster recovery protected workloads can be failed over or relocated when the active hub becomes unavailable.
- **Switching to passive hub cluster**
  Switch to the passive hub cluster in a Metro DR setup by restoring backups and verifying that managed clusters and DRPolicies are successfully imported when the active hub is down or unreachable.

Configuring passive hub cluster

Configure the passive hub cluster in a Metro DR environment so that disaster recovery protected workloads can be failed over or relocated when the active hub becomes unavailable.

About this task

To perform hub recovery in case the active hub is down or unreachable, follow the procedure in this section to configure the passive hub cluster and then failover or relocate the disaster recovery protected workloads.

Procedure

1. Ensure that RHACM operator and MultiClusterHub is installed on the passive hub cluster. See RHACM installation guide for instructions.
   After the operator is successfully installed, a popover with a message that the Web console update is available appears on the user interface. Click **Refresh web console** from this popover for the console changes to reflect.
2. Before hub recovery, configure backup and restore. See Backup and restore topics of *RHACM Business continuity* guide.
3. Install the multicluster orchestrator (MCO) operator along with Red Hat OpenShift GitOps operator on the passive RHACM hub prior to the restore. For instructions to restore your RHACM hub, see Installing Fusion Data Foundation Multicluster Orchestrator operator.
4. Ensure that `.spec.cleanupBeforeRestore` is set to `None` for the `Restore.cluster.open-cluster-management.io` resource. For details, see Restoring passive resources while checking for backups chapter of RHACM documentation.
5. If SSL access across clusters was configured manually during setup, then re-configure SSL access across clusters. For instructions, see Configuring SSL access across clusters chapter.
6. On the passive hub, add label to `openshift-operators` namespace to enable basic monitoring of `VolumeSyncronizationDelay` alert using this command. For alert details, see Disaster recovery alerts chapter.

Switching to passive hub cluster

Switch to the passive hub cluster in a Metro DR setup by restoring backups and verifying that managed clusters and DRPolicies are successfully imported when the active hub is down or unreachable.

About this task

Use this procedure when active hub is down or unreachable.

Procedure

1. Restore the backups on the passive hub cluster. For information, see Operations > Business continuity > Backup and restore > Restoring a backup in Red Hat Advanced Cluster Management for Kubernetes product documentation.

Important: Recovering a failed hub to its passive instance only restores applications and their DR protected state to its last scheduled backup. Any application that was DR protected after the last scheduled backup would need to be protected again on the new hub.

2. During the restore procedure, to avoid eviction of resources when `ManifestWorks` are not regenerated correctly, you can enlarge the `AppliedManifestWork eviction` grace period.
   a. Verify that the restore is complete.

   ```
   oc -n <restore-namespace> wait restore <restore-name> --for=jsonpath='{.status.phase}'=Finished --timeout=120s
   ```

   b. After the restore is completed, on the hub cluster, check for existing global `KlusterletConfig`
   - If global `KlusterletConfig` exists then edit and set the value for `appliedManifestWorkEvictionGracePeriod` parameter to a larger value. For example, 24 hours or more.
   - If global `KlusterletConfig` does not exist, then create the `Klusterletconfig` using the following yaml:

     ```
     apiVersion: config.open-cluster-management.io/v1alpha1
     kind: KlusterletConfig
     metadata:
       name: global
     spec:
       appliedManifestWorkEvictionGracePeriod: "24h"
     ```

     The configuration is propagated to all the managed clusters automatically.

3. Verify that the Primary and Seconday managed clusters are successfully imported into the RHACM console and they are accessible. If any of the managed clusters are down or unreachable then they are not successfully imported.
4. Wait until DRPolicy validation succeeds.
5. Verify that the **DRPolicy** is created successfully. Run this command on the **Hub cluster** for each of the DRPolicy resources created, where *<drpolicy_name>* is replaced with a unique name.

   ```
   oc get drpolicy <drpolicy_name> -o jsonpath='{.status.conditions[].reason}{"\n"}'
   ```

   Example output:

   ```
   Succeeded
   ```

6. Refresh the RHACM console to make the DR monitoring dashboard tab accessible if it was enabled on the Active hub cluster.
7. After all components are recovered, edit the global `KlusterletConfig` on the new hub and remove the parameter `appliedManifestWorkEvictionGracePeriod` and its value.
8. If only the active hub cluster is down, restore the hub by performing hub recovery, and restoring the backups on the passive hub. If the managed clusters are still accessible, no further action is required.
9. If the primary managed cluster is down, along with the active hub cluster, you need to fail over the workloads from the primary managed cluster to the secondary managed cluster. For failover instructions, based on your workload type, see Subscription-based application failover between managed clusters or ApplicationSet-based application failover between managed clusters.
10. Verify that the failover is successful. When the Primary managed cluster is down, then the PROGRESSION status for the workload would be in `Cleaning Up` phase until the down managed cluster is back online and successfully imported into the RHACM console.
    On the passive hub cluster, run the following command to check the PROGRESSION status.

    ```
    oc get drpc -o wide -A
    ```

    ```
    NAMESPACE      NAME                                AGE    PREFERREDCLUSTER    FAILOVERCLUSTER    DESIREDSTATE    CURRENTSTATE
    PROGRESSION    START TIME             DURATION       PEER READY
    busybox        busybox-cephfs-placement-1-drpc    42m    ocp4perf1                                                  Deployed
    Completed      2024-04-18T09:03:05Z   1.044537443s   True
    ```

Regional-DR configuration for Fusion Data Foundation

This section of the guide provides you with insights into regional disaster recovery (Regional-DR) configuration for Fusion Data Foundation to failover an application from one Red Hat® OpenShift® Container Platform cluster to another and then failback the same application to the original primary cluster.

In this case the OpenShift Container Platform clusters are created or imported using Red Hat Advanced Cluster Management (RHACM).

Though Regional-DR solution is built on Asynchronous data replication and hence could have a potential data loss but provides the protection against a broad set of failures.

This is a general overview of the Regional-DR steps required to configure and execute OpenShift Disaster Recovery (ODR) capabilities using Fusion Data Foundation and RHACM across two distinct OpenShift Container Platform clusters separated by distance. In addition to these two clusters called managed clusters, a third OpenShift Container Platform cluster is required to serve as the Red Hat Advanced Cluster Management (RHACM) hub cluster.

Note: Regional-DR is currently supported only on VMware, bare metal, and hybrid cloud (one cluster in the cloud) environments.
Important: You can now set up Regional disaster recovery solutions for workloads based on OpenShift virtualization technology using OpenShift Data Foundation. For more information, see the knowledgebase article.

- **Components of Regional-DR solution**
  Regional-DR is composed of Red Hat Advanced Cluster Management for Kubernetes, IBM Storage Ceph, and Fusion Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters.
- **Hybrid cloud Regional-DR solution**
  Pair an on-premise deployment with a secondary cluster in the cloud to enable hybrid disaster recovery and workload flexibility across bare metal, vSphere, Azure, and Azure Red Hat OpenShift environments.
- **Regional-DR deployment workflow**
  This section provides an overview of the steps required to configure and deploy Regional-DR capabilities using the latest versions of IBM Fusion Data Foundation across two distinct OpenShift Container Platform clusters. In addition to two managed clusters, a third OpenShift Container Platform cluster is required to deploy the Red Hat Advanced Cluster Management (RHACM).
- **Requirements for enabling Regional-DR**
  This section lists all the prerequisites for implementing the Disaster Recovery solution supported by IBM Fusion Data Foundation.

- **Creating a Fusion Data Foundation cluster on managed clusters**
  Create a Fusion Data Foundation cluster on each managed cluster to configure storage replication between two OpenShift Container Platform clusters for disaster recovery.
- **Installing Fusion Data Foundation Multicluster Orchestrator operator**
  Fusion Data Foundation Multicluster Orchestrator is a controller that is installed from the Software Catalog of Red Hat OpenShift Container Platform on the hub cluster.
- **Configuring SSL access across clusters**
  Configure network (SSL) access between the Primary and Secondary clusters so that metadata can be stored on the alternate cluster in a Multicloud Gateway (MCG) object bucket using a secure transport protocol and in the Hub cluster for verifying access to the object buckets.
- **Creating Disaster Recovery Policy on Hub cluster**
  The Disaster Recovery Policy (DRPolicy) resource specifies OpenShift Container Platform clusters participating in the disaster recovery solution. DRPolicy is a cluster scoped resource that users can apply to applications that require Disaster Recovery solution.
- **Disaster recovery protection for cloned and restored RBD volumes**
  The CephCSI driver does necessary operations on the cloned or restored RADOS Block Device (RBD) volumes to prepare them for disaster recovery protection when you enable the disaster recovery support for restored and cloned PersistentVolumeClaims.
- **Create sample application for testing disaster recovery solution**
  IBM Fusion Data Foundation disaster recovery (DR) solution supports disaster recovery for applications that are managed by Red Hat Advanced Cluster Management for Kubernetes (RHACM).
- **Virtual machine garbage collection automation**
  Virtual machine garbage collection automation provides an automated mechanism to clean up virtual machine resources on the source cluster during disaster recovery failover or relocation workflows, reducing manual effort and supporting lower recovery time objectives.
- **Subscription-based application failover between managed clusters**
  Fail over a subscription-based application from a primary managed cluster to a secondary managed cluster to maintain application availability during a disaster or cluster failure.
- **ApplicationSet-based application failover between managed clusters**
  Fail over an ApplicationSet-based application from a primary managed cluster to a secondary managed cluster to maintain application availability during a disaster or cluster failure.
- **Relocating a Subscription-based application between managed clusters**
  Relocate an application to its preferred location when all managed clusters are available.
- **Relocating an ApplicationSet-based application between managed clusters**
  Relocate an application to its preferred location when all managed clusters are available.
- **Disaster recovery protection for discovered applications**
  Disaster recovery protection for discovered applications extends IBM Fusion Data Foundation DR capabilities to workloads deployed directly on managed clusters without using Red Hat Advanced Cluster Management.
- **Recovering to a replacement cluster with Regional-DR**
  When there is a failure with the primary cluster, you get the options to either repair, wait for the recovery of the existing cluster, or replace the cluster entirely if the cluster is irredeemable. This solution guides you when replacing a failed primary cluster with a new cluster and enables failback (relocate) to this new cluster.
- **Viewing Recovery Point Objective values for disaster recovery enabled applications**
  Recovery Point Objective (RPO) value is the most recent sync time of persistent data from the cluster where the application is currently active to its peer. This sync time helps determine duration of data lost during failover.
- **Hub recovery using Red Hat Advanced Cluster Management**
  Hub recovery using Red Hat Advanced Cluster Management enables you to use a passive hub cluster to failover or relocate disaster recovery protected workloads when the active hub is unavailable.
- **Enabling granular disaster recovery for individual or groups of virtual machines in a namespace [Technology preview]**
  You can enable granular disaster recovery (DR) for individual virtual machines (VMs) or VM groups within the same namespace, allowing independent failover and relocation actions.

Components of Regional-DR solution

Regional-DR is composed of Red Hat Advanced Cluster Management for Kubernetes, IBM Storage Ceph, and Fusion Data Foundation components to provide application and data mobility across OpenShift Container Platform clusters.

Note: Regional-DR is supported with Fusion Data Foundation 4.17 and Red Hat Advanced Cluster Management for Kubernetes 2.11 combinations only.

Red Hat Advanced Cluster Management for Kubernetes

Red Hat Advanced Cluster Management (RHACM) provides the ability to manage multiple clusters and application lifecycles. Hence, it serves as a control plane in a multi-cluster environment.

RHACM is split into two parts:

RHACM Hub
: Components that run on the multi-cluster control plane.

Managed clusters
: Components that run on the clusters that are managed.

For more information about RHACM, see about this product, see About > Welcome to Red Hat Advanced Cluster Management for Kubernetes and Applications > Managing applications within Red Hat Advanced Cluster Management for Kubernetes product documentation.

Fusion Data Foundation

Fusion Data Foundation provides the ability to provision and manage storage for stateful applications in an OpenShift Container Platform cluster. It is backed by Ceph as the storage provider, whose lifecycle is managed by Rook in the Fusion Data Foundation component stack and Ceph-CSI provides the provisioning and management of Persistent Volumes for stateful applications.

Fusion Data Foundation is now enhanced with the following abilities for disaster recovery:

- Enable RBD block pools for mirroring across Fusion Data Foundation instances (clusters)
- Ability to mirror specific images within an RBD block pool
- Provides `csi-addons` to manage per Persistent Volume Claim (PVC) mirroring

OpenShift DR

OpenShift DR is a disaster recovery orchestrator for stateful applications across a set of peer OpenShift clusters, which are deployed and managed by using RHACM and provides cloud-native interfaces to orchestrate the life cycle of an application's state on Persistent Volumes. These include:

- Protecting an application and its state relationship across OpenShift clusters.
- Failing over an application and its state to a peer cluster.
- Relocate an application and its state to the previously deployed cluster.

OpenShift DR is split into three components:

IBM Fusion Data Foundation Multicluster Orchestrator
: Installed on the Hub cluster with RHACM, it orchestrates configuration and peering of Fusion Data Foundation clusters for Metro and Regional DR relationships.

IBM Fusion Data Foundation DR Hub Operator
: Automatically installed as part of IBM Fusion Data Foundation Multicluster Orchestrator installation on the hub cluster to orchestrate failover or relocation of DR enabled applications.

IBM Fusion Data Foundation DR Cluster Operator
: Automatically installed on each managed cluster that is part of a Metro and Regional DR relationship to manage the lifecycle of all PVCs of an application.

Hybrid cloud Regional-DR solution

Pair an on-premise deployment with a secondary cluster in the cloud to enable hybrid disaster recovery and workload flexibility across bare metal, vSphere, Azure, and Azure Red Hat OpenShift environments.

Fusion Data Foundation supports pairing an on-premise Fusion Data Foundation deployment with a secondary cluster in the cloud, enabling hybrid disaster recovery and workload flexibility. This capability allows on-prem environments such as Bare Metal and vSphere to integrate with cloud platforms Azure and Azure Red Hat OpenShift (ARO), expanding Fusion Data Foundation's support for hybrid deployment scenarios.

Regional-DR deployment workflow

This section provides an overview of the steps required to configure and deploy Regional-DR capabilities using the latest versions of IBM Fusion Data Foundation across two distinct OpenShift Container Platform clusters. In addition to two managed clusters, a third OpenShift Container Platform cluster is required to deploy the Red Hat Advanced Cluster Management (RHACM).

Procedure

To configure your infrastructure, perform the following steps:

1. Ensure requirements across the Hub, Primary and Secondary Openshift Container Platform clusters that are part of the DR solution are met.
   See Requirements for enabling Regional-DR.
2. Install Fusion Data Foundation operator and create a storage system on Primary and Secondary-managed clusters.
   See Creating a Fusion Data Foundation cluster on managed clusters.
3. Install the ODF Multicluster Orchestrator on the Hub cluster.
   See Installing Fusion Data Foundation Multicluster Orchestrator operator.
4. Configure SSL access between the Hub, Primary and Secondary clusters.
   See Configuring SSL access across clusters.
5. Create a DRPolicy resource for use with applications requiring DR protection across the Primary and Secondary clusters.
   See Creating Disaster Recovery Policy on Hub cluster.
   Note: There can be more than a single policy.
6. Testing your disaster recovery solution with:
   a. Subscription-based application:
      - Create sample applications. See Creating a Subscription-based application.
      - Test failover and relocate operations using the sample application between managed clusters. See Subscription-based application failover between managed clusters and Relocating a Subscription-based application between managed clusters.

   b. ApplicationSet-based application:
      - Create sample applications. See Creating an ApplicationSet-based application.
      - Test failover and relocate operations using the sample application between managed clusters. See ApplicationSet-based application failover between managed clusters and Relocating an ApplicationSet-based application between managed clusters.

   c. Discovered applications
      - Ensure all requirements mentioned in Prerequisites is addressed. See Prerequisites for disaster recovery protection of discovered applications.
      - Create a sample discovered application. See Creating a sample discovered application
      - Enroll the discovered application. See Enrolling a sample discovered application for disaster recovery protection
      - Test failover and relocate. See Failover disaster recovery protected discovered application and Relocate disaster recovery protected discovered application.

Requirements for enabling Regional-DR

This section lists all the prerequisites for implementing the Disaster Recovery solution supported by IBM Fusion Data Foundation.

- You must have the following Red Hat® OpenShift® clusters that have network reachability between them:
  - **Hub cluster** where Red Hat Advanced Cluster Management for Kubernetes (RHACM) operator is installed.
  - **Primary-managed cluster** where Fusion Data Foundation is running.
  - **Secondary-managed cluster** where Fusion Data Foundation is running.

  Note: For configuring hub recovery setup, you need a 4th cluster which acts as the passive hub. The primary managed cluster (Site-1) can be co-situated with the active RHACM hub cluster while the passive hub cluster is situated along with the secondary managed cluster (Site-2). Alternatively, the active RHACM hub cluster can be placed in a neutral site (Site-3) that is not impacted by the failures of either of the primary managed cluster at Site-1 or the secondary cluster at Site-2. In this situation, if a passive hub cluster is used it can be placed with the secondary cluster at Site-2. For more information, see Configuring passive hub cluster.
- Ensure that RHACM operator and Multicluster Hub is installed on the Hub cluster. For detailed instructions, see Installation and upgrade > Install in Red Hat Advanced Cluster Management for Kubernetes product documentation.
  After the operator is successfully installed, a popover with a message that the Web console update is available appears on the user interface. Click **Refresh web console** from this popover for the console changes to reflect.
  Important: Ensure that application traffic routing and redirection are configured appropriately.
- On the Hub cluster,
  - Go to All Clusters > Infrastructure > Clusters.
  - Import or create the **Primary-managed cluster** and the **Secondary-managed cluster** by using the RHACM console.
  - Choose the appropriate options for your environment.

  For instructions, see Creating clusters and Cluster import within the Cluster management and multicluster engine > Clusters guide in Red Hat Advanced Cluster Management for Kubernetes product documentation.
  Note: Regional-DR is supported only with Fusion Data Foundation 4.16 or higher and Red Hat Advanced Cluster Management for Kubernetes 2.10 or higher.
- Connect the private Red Hat OpenShift cluster and service networks using the RHACM Submariner add-ons. Verify that the two clusters have non-overlapping service and cluster private networks. Otherwise, ensure that the Globalnet is enabled during the Submariner add-ons installation.
  Run the following command for each of the managed clusters to determine if Globalnet needs to be enabled. The example shown here is for non-overlapping cluster and service networks so Globalnet would not be enabled.

```
oc get networks.config.openshift.io cluster -o json | jq .spec
```

```
Example output for Primary cluster:
```

```
{
  "clusterNetwork": [
    {
      "cidr": "10.5.0.0/16",
      "hostPrefix": 23
    }
  ],
  "externalIP": {
    "policy": {}
  },
  "networkType": "OVNKubernetes",
  "serviceNetwork": [
    "10.15.0.0/16"
  ]
}
```

```
Example output for Secondary cluster:
```

```
{
  "clusterNetwork": [
    {
      "cidr": "10.6.0.0/16",
      "hostPrefix": 23
    }
  ],
  "externalIP": {
    "policy": {}
  },
  "networkType": "OVNKubernetes",
  "serviceNetwork": [
    "10.16.0.0/16"
  ]
}
```

Additionally, if Submariner and Fusion Data Foundation are already installed on the managed clusters, use the Red Hat OpenShift Data Foundation command-line interface (CLI) tool to get additional information about the clusters. This information can determine the need for enabling Globalnet during submariner installation based on the cluster service and private networks. Download the Red Hat OpenShift Data Foundation CLI tool from the customer portal.

Run the following command on one of the two managed clusters, where *PeerManagedClusterName(ClusterID)* is the name of the peer Fusion Data Foundation cluster:

```
 odf get dr-prereq <PeerManagedClusterName(ClusterID)>
```

If submariner is not installed with Globalnet on clusters with overlapping services, the following output shows:

```
Info: Submariner is installed.
Info: Globalnet is required.
Info: Globalnet is not enabled.
```

Note: This requires Submariner to be uninstalled and then reinstalled with Globalnet enabled.
If submariner is installed with Globalnet on clusters with overlapping services, the following output shows:

```
Info: Submariner is installed.
Info: Globalnet is required.
Info: Globalnet is enabled.
```

If submariner is installed without Globalnet on clusters with non-overlapping services, the following output shows:

```
Info: Submariner is installed.
Info: Globalnet is not required.
Info: Globalnet is not enabled.
```

If submariner is installed with Globalnet on clusters with non-overlapping services, the following output shows:

```
Info: Submariner is installed.
Info: Globalnet is not required.
Info: Globalnet is enabled.
```

For more information, see Submariner documentation.

Creating a Fusion Data Foundation cluster on managed clusters

Create a Fusion Data Foundation cluster on each managed cluster to configure storage replication between two OpenShift Container Platform clusters for disaster recovery.

About this task

To configure storage replication between the two OpenShift Container Platform clusters, create an Fusion Data Foundation storage system after you install the Fusion Data Foundation operator.
Note: Refer to instructions in Installing Fusion Data Foundation specific to your infrastructure.

Procedure

1. Install and configure the latest Fusion Data Foundation cluster on each of the managed clusters.
   For Fusion Data Foundation configuration, see Configuring Fusion Data Foundation.
2. Validate the successful deployment of Fusion Data Foundation on each managed cluster with the following command:

   ```
   oc get storagecluster -n openshift-storage ocs-storagecluster -o jsonpath='{.status.phase}{"\n"}'
   ```

   For the Multicloud Gateway (MCG):

   ```
   oc get noobaa -n openshift-storage noobaa -o jsonpath='{.status.phase}{"\n"}'
   ```

   If the status result is `Ready` for both queries on the **Primary managed cluster** and the **Secondary-managed cluster**, then continue with the next step.
3. In the OpenShift Web Console, navigate to Installed Operators > > Storage System > ocs-storagecluster-storagesystem > Resources and verify that **Status** of `StorageCluster` is `Ready` and has a green tick mark next to it.
4. Optional: If Globalnet was enabled when Submariner was installed, then edit the StorageCluster after the OpenShift Data Foundation install finishes.
   For Globalnet networks, manually edit the `StorageCluster` YAML file to add the `clusterID` and set enabled to `true`. Replace `<clustername>` with your RHACM imported or newly created managed cluster name. Edit the `StorageCluster` on both the **Primary managed cluster** and the **Secondary managed cluster**.
   Warning: Do not make this change in the `StorageCluster` unless you enabled Globalnet when Submariner was installed.

   ```
   oc edit storagecluster -o yaml -n openshift-storage
   ```

   ```
   spec:
     network:
       multiClusterService:
         clusterID: <clustername>
         enabled: true
   ```

5. After the preceding changes are made,
   a. Wait for the OSD pods to restart and OSD services to be created.
   b. Wait for all MONS to failover.
   c. Ensure that the MONS and OSD services are exported.

   ```
   oc get serviceexport -n openshift-storage

   NAME              AGE
   rook-ceph-mon-d   4d14h
   rook-ceph-mon-e   4d14h
   rook-ceph-mon-f   4d14h
   rook-ceph-osd-0   4d14h
   rook-ceph-osd-1   4d14h
   rook-ceph-osd-2   4d14h
   ```

   d. Ensure that cluster is in a Ready state and cluster health has a green tick indicating Health ok. Verify using step 3.

Installing Fusion Data Foundation Multicluster Orchestrator operator

Fusion Data Foundation Multicluster Orchestrator is a controller that is installed from the Software Catalog of Red Hat® OpenShift® Container Platform on the hub cluster.

Before you begin

- Ensure that you created the following `CatalogSource` manifest on the hub cluster:

  ```
  apiVersion: operators.coreos.com/v1alpha1
  kind: CatalogSource
  metadata:
    name: isf-data-foundation-catalog
    namespace: openshift-marketplace
  spec:
    displayName: Data Foundation Catalog
  ```

```
  grpcPodConfig:
    affinity:
      nodeAffinity:
        preferredDuringSchedulingIgnoredDuringExecution:
          - preference:
              matchExpressions:
                - key: gpu.isf.ibm.com
                  operator: DoesNotExist
            weight: 50
    securityContextConfig: restricted
  icon:
    base64data: ""
    mediatype: ""
  image: icr.io/cpopen/isf-data-foundation-catalog:v4.21
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 60m
```

Note: Ensure that the version number you specify in the **image** field of the **CatalogSource** manifest matches the Fusion Data Foundation version installed on the managed cluster.

- Ensure that you are logged in to the Red Hat OpenShift Container Platform as an administrator.
- Ensure that `ibm-spectrum-fusion-ns` is selected in the Project field.

Procedure

1. Go to Ecosystem > Software Catalog and use the keyword filter to search for `Multicluster Orchestrator`.
2. Click the IBM-provided Multicluster Orchestrator tile.
3. Keep all default settings and click Install.
   Ensure that the operator resources are installed in **openshift-operators** project and available to all namespaces.
   Note: The **IBM Fusion Data Foundation Multicluster Orchestrator** also installs the **DR Hub Operator** on the RHACM hub cluster as a dependency.
4. Verify that the operator **Pods** are in a *Running* state.
   The DR Hub operator is also installed at the same time in **openshift-operators** namespace.

   ```
   oc get pods -n openshift-operators
   ```

   Example output:

   ```
   NAME                                       READY   UP-TO-DATE   AVAILABLE   AGE
   odf-multicluster-console-6845b795b9-blxrn  1/1     Running      0           4d20h
   odfmo-controller-manager-f9d9dfb59-jbrsd   1/1     Running      0           4d20h
   ramen-hub-operator-6fb887f885-fss4w        2/2     Running      0           4d20h
   ```

Configuring SSL access across clusters

Configure network (SSL) access between the Primary and Secondary clusters so that metadata can be stored on the alternate cluster in a Multicloud Gateway (MCG) object bucket using a secure transport protocol and in the Hub cluster for verifying access to the object buckets.

About this task

If all of your OpenShift clusters are deployed using a signed and valid set of certificates for your environment then this section can be skipped. Proceed to Creating Disaster Recovery Policy on Hub cluster.

Procedure

1. Extract the ingress certificate for the Primary-managed cluster and save the output to primary.crt.

   ```
   oc get cm default-ingress-cert -n openshift-config-managed -o jsonpath="{['data']['ca-bundle\.crt']}" > primary.crt
   ```

2. Extract the ingress certificate for the Secondary-managed cluster and save the output to secondary.crt.

   ```
   oc get cm default-ingress-cert -n openshift-config-managed -o jsonpath="{['data']['ca-bundle\.crt']}" > secondary.crt
   ```

3. Create a new ConfigMap file to hold the remote cluster's certificate bundle with filename cm-clusters-crt.yaml.
   Note: There could be more or less than three certificates for each cluster, as shown in this example file. Also, ensure that the certificate contents are correctly indented after you copy and paste from the primary.crt and secondary.crt files that were created before.

   ```
   apiVersion: v1
   data:
     ca-bundle.crt: |
       -----BEGIN CERTIFICATE-----
       <copy contents of cert1 from primary.crt here>
       -----END CERTIFICATE-----

       -----BEGIN CERTIFICATE-----
       <copy contents of cert2 from primary.crt here>
       -----END CERTIFICATE-----

       -----BEGIN CERTIFICATE-----
       <copy contents of cert3 primary.crt here>
       -----END CERTIFICATE----
   ```

```
    -----BEGIN CERTIFICATE-----
    <copy contents of cert1 from secondary.crt here>
    -----END CERTIFICATE-----

    -----BEGIN CERTIFICATE-----
    <copy contents of cert2 from secondary.crt here>
    -----END CERTIFICATE-----

    -----BEGIN CERTIFICATE-----
    <copy contents of cert3 from secondary.crt here>
    -----END CERTIFICATE-----
kind: ConfigMap
metadata:
  name: user-ca-bundle
  namespace: openshift-config
```

4. Create the ConfigMap on the Primary-managed cluster, Secondary-managed cluster, and the Hub cluster.

   ```
   oc create -f cm-clusters-crt.yaml
   ```

   Example output:

   ```
   configmap/user-ca-bundle created
   ```

5. Patch default proxy resource on the Primary-managed cluster, Secondary-managed cluster, and the Hub cluster.

   ```
   oc patch proxy cluster --type=merge  --patch='{"spec":{"trustedCA":{"name":"user-ca-bundle"}}}'
   ```

   Example output:

   ```
   proxy.config.openshift.io/cluster patched
   ```

Creating Disaster Recovery Policy on Hub cluster

The Disaster Recovery Policy (DRPolicy) resource specifies OpenShift Container Platform clusters participating in the disaster recovery solution. DRPolicy is a cluster scoped resource that users can apply to applications that require Disaster Recovery solution.

Before you begin

Ensure that there is a minimum set of two managed clusters.

About this task

The Fusion Data Foundation MultiCluster Orchestrator Operator facilitates the creation of each DRPolicy and the corresponding DRClusters through the Multicluster Web console.

Procedure

1. On the **OpenShift console**, navigate to All Clusters > Data Services > Disaster recovery.
2. On the Overview tab, click Create a disaster recovery policy or you can navigate to Policies tab and click Create DRPolicy.
3. Enter Policy name.
   Ensure that each DRPolicy has a unique name (for example: `ocp4bos1-ocp4bos2-5m`).
4. Select two clusters from the list of managed clusters with which this new policy is associated.
   Note: If you get an error message "OSDs not migrated" after selecting the clusters, then follow the instructions from knowledgebase article on Migration of existing OSD to the optimized OSD in OpenShift Data Foundation for Regional-DR cluster before proceeding with the next step.
5. Replication policy is automatically set to `Asynchronous` (async), based on the OpenShift clusters selected and a Sync schedule option becomes available.
6. Set Sync schedule.
   Important: For every desired replication interval a new **DRPolicy** must be created with a unique name (such as: `ocp4bos1-ocp4bos2-10m`). The same clusters can be selected but the **Sync schedule** can be configured with a different replication interval in minutes/hours/days. The minimum is one minute.
7. Optional: Expand Advanced settings and select the Enable disaster recovery support for restored and cloned PersistentVolumeClaims (For Data Foundation only) checkbox.
   For more information on disaster recovery protection for cloned and restored RBD volumes, see Disaster recovery protection for cloned and restored RBD volumes.
   Important: This option should only be used with discovered applications.
8. Click Create.
9. Verify that the DRPolicy is created successfully.
   Run this command on the Hub cluster for each of the DRPolicy resources created, where *<drpolicy_name>* is replaced with your unique name.

   ```
   oc get drpolicy <drpolicy_name> -o jsonpath='{.status.conditions[].reason}{"\n"}'
   ```

   Example output: `Succeeded`

   When a DRPolicy is created, along with it, two DRCluster resources are also created. It might take up to 10 minutes for all three resources to be validated and for the status to show as `Succeeded`.

   Note: Editing of `SchedulingInterval`, `ReplicationClassSelector`, `VolumeSnapshotClassSelector` and `DRClusters` field values are not supported in the DRPolicy.
10. Verify the object bucket access from the Hub cluster to both the **Primary-managed cluster** and the **Secondary-managed cluster**.
    a. Get the names of the DRClusters on the Hub cluster.

       ```
       oc get drclusters
       ```

Example output:

```
NAME         AGE
ocp4bos1   4m42s
ocp4bos2   4m42s
```

b. Check S3 access to each bucket created on **each** managed cluster.
Use the DRCluster validation command, where *<drcluster_name>* is replaced with your unique name.
Note: Editing of `Region` and `S3ProfileName` field values are non supported in DRClusters.

```
oc get drcluster <drcluster_name> -o jsonpath='{.status.conditions[2].reason}{"\n"}'
```

Example output: `Succeeded`

Note: Ensure to run command for both DRClusters on the Hub cluster.

11. Verify that the IBM Fusion Data Foundation DR Cluster operator installation was successful on the **Primary-managed cluster** and the **Secondary-managed cluster**.

```
oc get csv,pod -n openshift-dr-system
```

Example output:

```
NAME                                                                  DISPLAY                          VERSION
REPLACES   PHASE
clusterserviceversion.operators.coreos.com/odr-cluster-operator.v4.14.0          Openshift DR Cluster Operator   4.14.0
Succeeded
clusterserviceversion.operators.coreos.com/volsync-product.v0.8.0                VolSync                         0.8.0
Succeeded

NAME                                          READY   STATUS    RESTARTS   AGE
pod/ramen-dr-cluster-operator-6467cf5d4c-cc8kz   2/2     Running   0          3d12h
```

You can also verify that IBM Fusion Data Foundation DR Cluster Operator is installed successfully on the **OperatorHub** of each managed cluster.

Note: On the initial run, VolSync operator is installed automatically. VolSync is used to set up volume replication between two clusters to protect CephFs-based PVCs. The replication feature is enabled by default.

12. Verify that the status of the IBM Fusion Data Foundation mirroring `daemon` health on the **Primary managed cluster** and the **Secondary managed cluster**.

```
oc get cephblockpool ocs-storagecluster-cephblockpool -n openshift-storage -o jsonpath='{.status.mirroringStatus.summary}
{"\n"}'
```

Example output:

```
{"daemon_health":"OK","health":"OK","image_health":"OK","states":{}}
```

CAUTION:
It could take up to 10 minutes for the daemon_health and health to go from Warning to OK. If the status does not become OK eventually, then use the RHACM console to verify that the Submariner connection between managed clusters is still in a healthy state. Do not proceed until all values are OK.

Disaster recovery protection for cloned and restored RBD volumes

The CephCSI driver does necessary operations on the cloned or restored RADOS Block Device (RBD) volumes to prepare them for disaster recovery protection when you enable the disaster recovery support for restored and cloned PersistentVolumeClaims.

Cloned and restored RBD volumes are copy-on-write clones of the corresponding data sources. Such RBD volumes are still linked to the parent volume to optimize space consumption and Ceph cluster performance. To mirror RBD images that have such volumes, the cloned and restored RBD volumes need to be separated from the parent RBD image.

Selecting the Enable disaster recovery support for restored and cloned PersistentVolumeClaims (For Data Foundation only) checkbox while Creating Disaster Recovery Policy on Hub cluster instructs CephCSI driver to do necessary operations on the cloned or restored RBD volumes to prepare them for disaster recovery protection. The necessary operation has the following consequences:

- Space consumption may increase up to the total capacity of all cloned and restored RBD volumes that are protected. Ensure that enough space available in the cluster so that the cluster does not full.
- Resource usage of the Ceph cluster increases.
- The time taken for the operation to complete depends on the number of the RBD volumes, size of each volume, Ceph cluster resource availability, and client IO going on the RBD volume. Therefore, the amount of time that is required for completion of the operation cannot be determined.

Create sample application for testing disaster recovery solution

IBM Fusion Data Foundation disaster recovery (DR) solution supports disaster recovery for applications that are managed by Red Hat® Advanced Cluster Management for Kubernetes (RHACM).

The following section detail how to create an application and apply a DRPolicy to an application:

- ApplicationSet-based applications
  OpenShift users that do not have cluster-admin permissions cannot create ApplicationSet-based applications. For more information on ApplicationSet-based application, see ApplicationSet in *Red Hat Advanced Cluster Management for Kubernetes product documentation*.

Note: Regional-DR now supports leveraging RBD and CephFS volumes using non-default replica-2 storage classes that are managed by Fusion Data Foundation. Support has also been extended to compression-enabled storage classes.

Multi-volume consistency for disaster recovery

By default, Fusion Data Foundation Disaster Recovery (DR) provides crash-consistent multi-volume consistency groups for Regional-DR to be used by applications that are deployed across multiple volumes.

Note: If you have upgraded from a previous Fusion Data Foundation version to 4.21, to enable consistency groups you must disable Regional-DR after the upgrade, and then re-enable Regional-DR.

- **Deleting a sample application**
  The sample application `busybox` can be deleted by using the Red Hat Advanced Cluster Management for Kubernetes (RHACM) console.

Creating an ApplicationSet-based application

Create an ApplicationSet-based application on the Hub cluster using the Argo CD Pull model to test Regional-DR failover and relocate across managed clusters.

Before you begin

- Ensure that the Red Hat OpenShift GitOps operator is installed on all three clusters: **Hub cluster**, **Primary managed cluster** and **Secondary managed cluster**. For instructions, see Installing Red Hat OpenShift GitOps Operator in web console.
- On the Hub cluster, ensure that both Primary and Secondary managed clusters are registered to GitOps. For registration instructions, see Application management and GitOps > GitOps > Registering managed clusters to Red Hat OpenShift GitOps operator in Red Hat Advanced Cluster Management for Kubernetes product documentation. Then check if the Placement used by `GitOpsCluster` resource to register both managed clusters, has the tolerations to deal with cluster unavailability. You can verify if the following tolerations are added to the Placement using the command.

  ```
  oc get placement <placement-name> -n openshift-gitops -o yaml.
  ```

  ```
    tolerations:
    - key: cluster.open-cluster-management.io/unreachable
      operator: Exists
    - key: cluster.open-cluster-management.io/unavailable
      operator: Exists
  ```

  In case the tolerations are not added, see Application management and GitOps > GitOps in Red Hat Advanced Cluster Management for Kubernetes product documentation.
- Ensure that you have created the `ClusterRoleBinding` yaml on both the **Primary** and **Secondary managed** clusters. For instruction, see Application management and GitOps > GitOps > Deploying Argo CD with Push and Pull model > Prerequisites in Red Hat Advanced Cluster Management for Kubernetes product documentation.

Procedure

1. On the Hub cluster, navigate to All Clusters > Applications > Create application.
2. Select type as Argo CD ApplicationSet - Pull model.
3. In General step, enter your Application set name.
4. Select Argo server `openshift-gitops` and Requeue time as `180` seconds.
5. Click Next.
6. In the Repository location for resources section, select Repository type `Git`.
7. Enter the Git repository URL for the sample application, the GitHub Branch, and Path where the resources `busybox` Pod and PVC are created.
   a. Use the sample application repository as https://github.com/red-hat-storage/ocm-ramen-samples
   b. Select Revision as `main`.
   c. Choose one the following Path:
      - `workloads/deployment/odr-regional-rbd` to use RBD Regional-DR.
      - `workloads/deployment/odr-regional-cephfs` to use CephFS Regional-DR.
8. Enter Remote namespace value. (example, busybox-sample) and click Next.
9. Select Sync policy settings and click Next.
   You can choose one or more options.
10. Add a label *<name>* with its value set to the managed cluster name.
11. Click Next.
12. Review the setting details and click Submit.

Applying DRpolicy to an ApplicationSet-based application

Apply a disaster recovery policy to an ApplicationSet-based application from the Hub cluster console to enable volume replication and protect the application against cluster failure.

Before you begin

Ensure that both managed clusters referenced in the DRPolicy are reachable. If not, the application is not DR protected until both clusters are online.

Procedure

1. On the Hub cluster, navigate to All Clusters > Applications.
2. Click the Actions menu at the end of application to view the list of available actions.
3. Click Manage disaster recovery.

4. Click Enroll application.
5. Select Policy name and click Next.
6. Select an Application resource and then use PVC label selector to select **`PCV label`** for the selected resource.
   Note: You can select more than one PVC label for the selected application resources.
7. After adding all the application resources, click Next.
8. In the Enroll managed application modal, review the policy configuration details and click Assign. The newly assigned Data policy details are displayed on the Manage disaster recovery modal.
9. Verify that you can view the assigned policy details on the Applications page.
   a. On the Applications page, navigate to the DR Status column and view the status.
      The status can be any of the following:

      - Protecting: Shown during initial DR setup before data synchronization begins
      - Healthy: All volumes and Kubernetes resources are syncing normally
      - Warning: Volumes or Kubernetes resources are syncing slower than usual
      - Critical: Volumes or Kubernetes resources are not syncing
      - Protection Error: An error occurred during application protection setup

      You can click the status to view the last synced time for application volumes and the DR policy assigned.

      Note: It may take a few minutes for the DR Status to move from Protecting to Healthy after disaster recovery is enabled. The DR status shows Protecting during initial setup, then automatically transition to health-based statuses (Healthy/Warning/Critical) once data synchronization begins.
   b. Failover and relocate status can be viewed in the DR Status column when initiated. These statuses can be clicked to view more details about the action. DR Status shows either Protecting, Healthy, Warning, or Critical when failover or relocate is not taking place. During failover or relocate operations, the status shows the operation-specific status (FailingOver, Relocating, and so on).

10. In the user interface, verify the volume consistency groups:
    a. In the Overview tab, at the end of the protected application row from the action menu (⋮), select Manage disaster recovery.
    b. In the modal, verify Volume group replication is *Enabled*.
    c. In the same modal, select View volume groups.
    d. Verify the volume consistency groups have the correct volumes and a *Synced* status.
11. Optional: For Rados block device (RBD), verify **`volumereplication`**, **`volumereplicationgroup`**, and **`volumegroupreplication`** on the primary cluster.
    For **`volumereplication`**:

    ```
    oc get volumereplication -A
    ```

    Example output:

    ```
    NAMESPACE        NAME                                        AGE    VOLUMEREPLICATIONCLASS                  SOURCEKIND
    SOURCENAME                                                      DESIREDSTATE   CURRENTSTATE
    busybox-sample   vr-bc4a166e-13bc-45cd-99f0-8bd516f7ffcc   112s   rbd-volumereplicationclass-1625360775
    VolumeGroupReplication   vgr-07a0050bee1f25720f5cae8d5daa6230-busybox-placement-drpc   primary        Primary
    ```

    For **`volumereplicationgroup`**:

    ```
    oc get volumereplicationgroup -A
    ```

    Example output:

    ```
    NAMESPACE        NAME                      DESIREDSTATE   CURRENTSTATE
    busybox-sample   busybox-placement-drpc   primary        Primary
    ```

    For **`volumegroupreplication`**:

    ```
    oc get volumegroupreplication -A
    ```

    Example output:

    ```
    NAMESPACE        NAME                                                                VOLUMEGROUPREPLICATIONCLASS
    VOLUMEGROUPREPLICATIONCONTENT                      DESIREDSTATE   CURRENTSTATE   AGE
    busybox-sample   vgr-07a0050bee1f25720f5cae8d5daa6230-busybox-placement-drpc   rbd-volumegroupreplicationclass-1625360775-
    1602718344   vgrcontent-bc4a166e-13bc-45cd-99f0-8bd516f7ffcc   primary        Primary        114s
    ```

12. Optional: For CephFS, verify **`ReplicationGroupSource`** and **`ReplicationSource`** on the Primary cluster.
    For **`ReplicationGroupSource`**:

    ```
    oc get ReplicationGroupSource -A
    ```

    Example output:

    ```
    NAMESPACE        NAME                                                LAST SYNC              DURATION          NEXT SYNC
    SOURCE                                                                                                        LAST SYNC
    START
    busybox-cephfs   busybox-cephfs-bbd241b65b6589782fddaa82636e358d   2026-03-18T11:20:14Z   1m16.462467415s   2026-03-
    18T11:25:00Z   {"matchLabels":{"ramendr.openshift.io/consistency-group":"busybox-cephfs-
    bbd241b65b6589782fddaa82636e358d"}}
    ```

    For **`ReplicationSource`**:

    ```
    oc get ReplicationSource -A
    ```

    Example output:

    ```
    NAMESPACE        NAME          SOURCE           LAST SYNC              DURATION        NEXT SYNC
    busybox-cephfs   busybox-pvc   vs-busybox-pvc   2026-03-18T11:20:14Z   16.450725976s
    ```

13. Optional: For CephFS, verify **`ReplicationGroupDestination`** and **`ReplicationDestination`** on the Secondary cluster.
    For **`ReplicationGroupDestination`**:

    ```
    oc get ReplicationGroupDestination -A
    ```

    Example output:

```
NAMESPACE         NAME                                                LAST SYNC              DURATION          LAST SYNC START
busybox-cephfs    busybox-cephfs-b1f2b5bac27b3d8805437424895d212c      2026-03-18T11:20:15Z   47.700328282s     2026-03-
18T11:20:15Z
```

For `ReplicationDestination`:

```
oc get ReplicationDestination -A
```

Example output:

```
NAMESPACE         NAME           LAST SYNC              DURATION          NEXT SYNC
busybox-cephfs    busybox-pvc    2026-03-18T11:20:15Z   47.401909636s
```

Deleting a sample application

The sample application `busybox` can be deleted by using the Red Hat Advanced Cluster Management for Kubernetes (RHACM) console.

Before you begin

Important: When deleting a DR protected application, access to both clusters that belong to the DRPolicy is required. This is to ensure that all protected API resources and resources in the respective S3 stores are cleaned up as part of removing the DR protection. If access to one of the clusters is not healthy, deleting the `DRPlacementControl` resource for the application, on the hub, would remain in the `Deleting` state.
Do not begin deleting a sample application until the failover and relocate testing is completed and the application is ready to be removed from RHACM and the managed clusters.

About this task

Procedure

1. On the **Hub cluster**, navigate to Applications.
2. Search for the sample application to be deleted (for example, `busybox`).
3. Click the Action menu next to the application that you want to delete.
4. Click Delete application.
   When the Delete application is selected a new screen appears asking if the application-related resources should also be deleted.
5. Select Remove application related resources to delete the Subscription and PlacementRule.
6. Click Delete.
   This deletes the `busybox` application on the Primary-managed cluster (or the cluster that the application was running on).
7. In addition to the resources deleted by using the RHACM console, delete the `DRPlacementControl` if it is not auto-deleted after deleting the `busybox` application.
   a. Log in to the Red Hat OpenShift web console for the Hub cluster and navigate to Installed Operators for the project `busybox-sample`.
      For ApplicationSet applications, select the project as `openshift-gitops`.
   b. Click OpenShift DR Hub Operator > DRPlacementControl.
   c. Click the Action menu next to the `busybox` application DRPlacementControl that you want to delete.
   d. Click Delete DRPlacementControl.
   e. Click Delete.

   Note: This process can be used to delete any application with a `DRPlacementControl` resource.

Virtual machine garbage collection automation

Virtual machine garbage collection automation provides an automated mechanism to clean up virtual machine resources on the source cluster during disaster recovery failover or relocation workflows, reducing manual effort and supporting lower recovery time objectives.

Overview

Virtual machine (VM) garbage collection automation provides an automated mechanism to clean up VM resources on the source cluster during Disaster Recovery (DR) failover or relocation workflows. This reduces manual effort and helps users perform DR operations more quickly, supporting lower Recovery Time Objectives (RTO).

Prerequisites

- Virtual machine protection must be enabled through the Fusion Data Foundation Virtualization console.

How it works

During DR failover or relocation, VM resources that are protected and orchestrated by the Fusion Data Foundation Virtualization console are automatically deleted from the source cluster.

This deletion uses Kubernetes' owner-based garbage collection principles, ensuring that associated child resources, such as disks and pods, are also removed cleanly as part of normal Kubernetes garbage collection.

Known risks and limitations

- Automation applies only to VM resources whose protection is managed by the Fusion Data Foundation Virtualization console.
- Other discovered application workload types, including GitOps-managed and other non-VM resources, remain out of scope and require manual cleanup.
- Automation is limited to resources directly owned by the VM lifecycle.

Subscription-based application failover between managed clusters

Fail over a subscription-based application from a primary managed cluster to a secondary managed cluster to maintain application availability during a disaster or cluster failure.

Before you begin

- If your setup has active and passive RHACM hub clusters, see Hub recovery using Red Hat Advanced Cluster Management.
- When primary cluster is in a state other than Ready, check the actual status of the cluster as it might take some time to update.
  1. Go to RHACM console > Infrastructure > Clusters > Cluster list tab.
  2. Check the status of both the managed clusters individually before performing a failover operation.

  However, failover operation can still be run when the cluster you are failing over to is in a *Ready* state.
- Run the following command on the Hub Cluster to check if `lastGroupSyncTime` is within an acceptable data loss window, when compared to current time.

  ```
  oc get drpc -o yaml -A | grep lastGroupSyncTime
  ```

  Example output:

  ```
  [...]
  lastGroupSyncTime: "2023-07-10T12:40:10Z"
  ```

About this task

Failover is a process that transitions an application from a primary cluster to a secondary cluster in the event of a primary cluster failure. While failover provides the ability for the application to run on the secondary cluster with minimal interruption, making an uninformed failover decision can have adverse consequences, such as complete data loss in the event of unnoticed replication failure from primary to secondary cluster. If a significant amount of time has gone by since the last successful replication, it's best to wait until the failed primary is recovered.

`LastGroupSyncTime` is a critical metric that reflects the time since the last successful replication occurred for all PVCs associated with an application. In essence, it measures the synchronization health between the primary and secondary clusters. So, prior to initiating a failover from one cluster to another, check for this metric and only initiate the failover if the **`LastGroupSyncTime`** is within a reasonable time in the past.

Note: During the course of failover the Ceph-RBD mirror deployment on the failover cluster is scaled down to ensure a clean failover for volumes that are backed by Ceph-RBD as the storage provisioner.

Procedure

1. Failover the protected application using either of the following paths:
   - **Option 1**: From the Applications page

     On the Hub cluster, navigate to Applications. Click Actions (⋮) and select Failover application.

   - **Option 2**: From the Protected Applications page

     On the Hub cluster, navigate to Data Services > Disater recovery > Protected Applications > Actions (⋮) > Failover.
2. After the Failover application popup is shown, select Policy and Target cluster to which the associated application fails over in a disaster.
3. Click the Select subscription group dropdown to verify the default selection or modify this setting.
   By default, the subscription group that replicates for the application resources is selected.
4. Check the status of the Failover readiness.
   - If the status is *Ready* with a green tick, it indicates that the target cluster is ready for failover to start. Proceed to step 5.
   - If the status is *Unknown* or *Not ready*, then wait until the status changes to *Ready*.

   Important:
   If there are data inconsistencies caused by synchronization delays, a warning message appears stating **Inconsistent data on target cluster**. This alerts to the possibility of data loss if the failover is initiated. The message is no longer displayed when data synchronization is complete.
5. Click Initiate.
   All the system workloads and their available resources are now transferred to the target cluster.
6. In the DR status column, you can view the progress of the failover. The modal shows the progress of the failover as Preparing > Failover > Restoring > Clean up.
7. Verify that the activity status shows as *FailedOver* for the application.
   a. Go to the Applications > Overview tab.
   b. In the Data policy column, click the policy link for the application you applied the policy to.
   c. On the Data Policy popover page, click the View more details link.
   d. Verify that you can see one or more policy names and the ongoing activities (Last sync time and Activity status) associated with the policy in use with the application.

ApplicationSet-based application failover between managed clusters

Fail over an ApplicationSet-based application from a primary managed cluster to a secondary managed cluster to maintain application availability during a disaster or cluster failure.

Before you begin

- If your setup has active and passive RHACM hub clusters, see Hub recovery using Red Hat Advanced Cluster Management.
- When primary cluster is in a state other than Ready, check the actual status of the cluster as it might take some time to update.
  1. Go to RHACM console > Infrastructure > Clusters > Cluster list tab.
  2. Check the status of both the managed clusters individually before performing a failover operation.

  However, failover operation can still be run when the cluster you are failing over to is in a *Ready* state.

- Run the following command on the Hub Cluster to check if `lastGroupSyncTime` is within an acceptable data loss window, when compared to current time.

  ```
  oc get drpc -o yaml -A | grep lastGroupSyncTime
  ```

  Example output:

  ```
  [...]
  lastGroupSyncTime: "2023-07-10T12:40:10Z"
  ```

About this task

Failover is a process that transitions an application from a primary cluster to a secondary cluster in the event of a primary cluster failure. While failover provides the ability for the application to run on the secondary cluster with minimal interruption, making an uninformed failover decision can have adverse consequences, such as complete data loss in the event of unnoticed replication failure from primary to secondary cluster. If a significant amount of time has gone by since the last successful replication, it's best to wait until the failed primary is recovered.

`LastGroupSyncTime` is a critical metric that reflects the time since the last successful replication occurred for all PVCs associated with an application. In essence, it measures the synchronization health between the primary and secondary clusters. So, prior to initiating a failover from one cluster to another, check for this metric and only initiate the failover if the `LastGroupSyncTime` is within a reasonable time in the past.

Note: During the course of failover the Ceph-RBD mirror deployment on the failover cluster is scaled down to ensure a clean failover for volumes that are backed by Ceph-RBD as the storage provisioner.

Procedure

1. Failover the protected application using either of the following paths:
   - **Option 1**: From the Applications page

     On the Hub cluster, navigate to Applications. Click Actions (⋮) and select Failover application.

   - **Option 2**: From the Protected Applications page

     On the Hub cluster, navigate to Data Services > Disaster recovery > Protected Applications > Actions (⋮) > Failover.

2. When the Failover application modal is shown, verify the details that are presented are correct and check the status of the Failover readiness. If the status is *Ready* with a green tick, it indicates that the target cluster is ready for failover to start.
   Important:
   If there are data inconsistencies caused by synchronization delays, a warning message appears stating **Inconsistent data on target cluster**. This alerts to the possibility of data loss if the failover is initiated. The message is no longer displayed when data synchronization is complete.

3. Click Initiate.
   All the system workloads and their available resources are now transferred to the target cluster.
4. In the DR status column, you can view the progress of the failover. The modal shows the progress of the failover as Preparing > Failover > Restoring > Clean up.
5. Verify that the activity status shows as *FailedOver* for the application.
   a. Go to the Applications > Overview tab.
   b. In the Data policy column, click the policy link for the application you applied the policy to.
   c. On the Data Policy popover page, verify that you can see one or more policy names and the ongoing activities associated with the policy in use with the application.

Relocating a Subscription-based application between managed clusters

Relocate an application to its preferred location when all managed clusters are available.

Before you begin

- If your setup has active and passive RHACM hub clusters, see Hub recovery using Red Hat Advanced Cluster Management.
- When the primary cluster is in a state other than Ready, check the actual status of the cluster as it might take some time to update. Relocate can be performed only when both primary and preferred clusters are up and running.
  1. Go to RHACM console > Infrastructure > Clusters > Cluster list tab.
  2. Check the status of both the managed clusters individually before performing a relocate operation.
- Perform relocate when `lastGroupSyncTime` is within the replication interval (for example, 5 minutes) when compared to current time. This is recommended to minimize the Recovery Time Objective (RTO) for any single application.
  Run this command on the Hub Cluster:

  ```
  oc get drpc -o yaml -A | grep lastGroupSyncTime
  ```

  Example output:

  ```
  [...]
  lastGroupSyncTime: "2023-07-10T12:40:10Z"
  ```

Compare the output time (UTC) to current time to validate that all **lastGroupSyncTime** values are within their application replication interval. If not, wait to Relocate until this is true for all **lastGroupSyncTime** values.

Procedure

1. Relocate the protected application using either of the following paths:

   Option 1: From the Applications page
   a. On the Hub cluster, navigate to All Clusters > Applications.
   b. In the Overview tab, at the end of the application row from the action menu **(⋮)**, select Manage diaster recovery.
   c. Click Relocate.

   Option 2: From the Protected Applications page
   - On the Hub cluster, navigate to Data Services > Disater recovery > Protected Applications > Actions (⋮) > Relocate.

2. When the Relocate application modal is shown, select Policy and Target cluster to which the associated application relocates to in case of a disaster.
3. By default, the subscription group that deploys the application resources is selected. Click the Select subscription group dropdown to verify the default selection or modify this setting.
4. Check the status of the Relocation readiness.
   - If the status is *Ready* with a green tick, it indicates that the target cluster is ready for failover to start. Proceed to step 5.
   - If the status is *Unknown* or *Not ready*, then wait until the status changes to *Ready*.

   Important:
   If there are data inconsistencies caused by synchronization delays, a warning message appears stating **Inconsistent data on target cluster**. This alerts to the possibility of data loss if the failover is initiated. The message is no longer displayed when data synchronization is complete.

5. Click Initiate.
   All the system workloads and their available resources are now transferred to the target cluster.
6. In the DR status column, you can view the progress of the failover. The modal shows the progress of the failover as Preparing > Syncing > Restoring > Clean up.
7. Verify that the activity status shows as *Relocated* for the application.
   a. Go to the Applications > Overview tab.
   b. In the Data policy column, click the policy link for the application you applied the policy to.
   c. On the Data Policy popover page, click the View more details link.
   d. Verify that you can see one or more policy names and the ongoing activities (Last sync time and Activity status) associated with the policy in use with the application.

Relocating an ApplicationSet-based application between managed clusters

Relocate an application to its preferred location when all managed clusters are available.

Before you begin

- If your setup has active and passive RHACM hub clusters, see Hub recovery using Red Hat Advanced Cluster Management.
- When primary cluster is in a state other than Ready, check the actual status of the cluster as it might take some time to update. Relocate can only be performed when both primary and preferred clusters are up and running.
  1. Go to RHACM console > Infrastructure > Clusters > Cluster list tab.
  2. Check the status of both the managed clusters individually before performing a relocate operation.
- Perform relocate when **lastGroupSyncTime** is within the replication interval (for example, 5 minutes) when compared to current time. This is recommended to minimize the Recovery Time Objective (RTO) for any single application.
  Run this command on the Hub Cluster:

  ```
  oc get drpc -o yaml -A | grep lastGroupSyncTime
  ```

  Example output:

  ```
  [...]
  lastGroupSyncTime: "2023-07-10T12:40:10Z"
  ```

  Compare the output time (UTC) to current time to validate that all **lastGroupSyncTime** values are within their application replication interval. If not, wait to Relocate until this is true for all **lastGroupSyncTime** values.

Procedure

1. Relocate the protected application using either of the following paths:

   Option 1: From the Applications page
   a. On the Hub cluster, navigate to All Clusters > Applications.
   b. In the Overview tab, at the end of the application row from the action menu **(⋮)**, select Manage diaster recovery.
   c. Click Relocate.

   Option 2: From the Protected Applications page
   - On the Hub cluster, navigate to Data Services > Disater recovery > Protected Applications > Actions (⋮) > Relocate.

2. When the Relocate application modal is shown, select Policy and Target cluster to which the associated application relocates to in case of a disaster.
   Important:
   If there are data inconsistencies caused by synchronization delays, a warning message appears stating **Inconsistent data on target cluster**. This alerts to the possibility of data loss if the failover is initiated. The message is no longer displayed when data synchronization is complete.

3. Click Initiate.

All the system workloads and their available resources are now transferred to the target cluster.

4. Close the modal window and track the status using the Data policy column on the Applications page.
   The status moves from Preparing > Syncing > Restoring > Clean up.
5. Verify that the activity status shows as *Relocated* for the application.
   a. Go to Applications > Overview.
   b. In the Data policy column, click the policy link for the application you applied the policy to.
   c. On the Data Policy popover page, verify that you can see one or more policy names and the relocation status associated with the policy in use with the application.

Disaster recovery protection for discovered applications

Disaster recovery protection for discovered applications extends IBM Fusion Data Foundation DR capabilities to workloads deployed directly on managed clusters without using Red Hat Advanced Cluster Management.

IBM Fusion Data Foundation now provides disaster recovery (DR) protection and support for workloads that are deployed in one of the managed clusters directly without using Red Hat Advanced Cluster Management (RHACM). These workloads are called discovered applications.

The workloads that are deployed using RHACM are now called managed applications. When a workload is deployed directly on one of the managed clusters without using RHACM, then those workloads are called discovered applications. Though these workload details can be seen on the RHACM console, the application lifecycle (create, delete, edit) is not managed by RHACM.

- **Prerequisites for disaster recovery protection of discovered applications**
  This section provides instructions to guide you through the prerequisites for protecting discovered applications. This includes tasks such as assigning a data policy and initiating DR actions such as failover and relocate.
- **Creating a sample discovered application**
  To test failover from the **Primary managed cluster** to the **Secondary managed cluster** and relocate for discovered applications, you need a sample application that is installed without using the RHACM create application capability.
- **Enrolling a sample discovered application for disaster recovery protection**
  This section guides you on how to apply an existing DR Policy to a discovered application from the Protected applications tab.
- **Discovered application failover and relocate**
  Discovered application failover and relocate follows a similar process to managed applications but requires additional manual steps because RHACM does not manage the lifecycle of discovered applications.
- **Disable disaster recovery for protected applications**
  This section guides you to disable disaster recovery resources when you want to delete the protected applications or when the application no longer needs to be protected.

Prerequisites for disaster recovery protection of discovered applications

This section provides instructions to guide you through the prerequisites for protecting discovered applications. This includes tasks such as assigning a data policy and initiating DR actions such as failover and relocate.

1. Ensure that all the DR configurations have been installed on the Primary managed cluster and the Secondary managed cluster.
2. Install the OADP 1.4 operator.
   Note: Any version before OADP 1.4 does not work for protecting discovered applications.
   a. On the **Primary** and **Secondary managed cluster**, navigate to OperatorHub and use the keyword filter to search for `OADP`.
   b. Click the OADP tile.
   c. Keep all default settings and click Install. Ensure that the operator resources are installed in the `openshift-adp` project.
      Note: If OADP 1.4 is installed after DR configuration has been completed then the `ramen-dr-cluster-operator` pods on the **Primary managed cluster** and the **Secondary managed cluster** in namespace `openshift-dr-system` must be restarted (deleted and recreated).
3. [Optional] Add CACertificates to `ramen-hub-operator-config` **ConfigMap**.
   Configure network (SSL) access between the primary and secondary clusters so that metadata can be stored on the alternate cluster in a Multicloud Gateway (MCG) object bucket using a secure transport protocol and in the Hub cluster for verifying access to the object buckets.
   Note: If all of your OpenShift clusters are deployed using a signed and valid set of certificates for your environment then this section can be skipped.
   If you are using self-signed certificates, then you have already created a **ConfigMap** named `user-ca-bundle` in the `openshift-config` namespace and added this ConfigMap to the default Proxy cluster resource.
   a. Find the encoded value for the CACertificates.

   ```
   oc get configmap user-ca-bundle -n openshift-config -o jsonpath="{['data']['ca-bundle\.crt']}" |base64 -w 0
   ```

   b. Add this base64 encoded value to the configmap `ramen-hub-operator-config` on the Hub cluster. The following example shows where to add CACertificates.

   ```
   oc edit configmap ramen-hub-operator-config -n openshift-operators
   ```

   ```
   [...]
       ramenOpsNamespace: openshift-dr-ops
       s3StoreProfiles:
       - s3Bucket: odrbucket-36bceb61c09c
         s3CompatibleEndpoint: https://s3-openshift-storage.apps.hyper3.vmw.ibmfusion.eu
         s3ProfileName: s3profile-hyper3-ocs-storagecluster
         s3Region: noobaa
         s3SecretRef:
           name: 60f2ea6069e168346d5ad0e0b5faa59bb74946f
         caCertificates: {input base64 encoded value here}
       - s3Bucket: odrbucket-36bceb61c09c
         s3CompatibleEndpoint: https://s3-openshift-storage.apps.hyper4.vmw.ibmfusion.eu
         s3ProfileName: s3profile-hyper4-ocs-storagecluster
         s3Region: noobaa
         s3SecretRef:
   ```

```
        name: cc237eba032ad5c422fb939684eb633822d7900
      caCertificates: {input base64 encoded value here}
```

4. Verify that there are DR secrets created in the OADP operator default namespace `openshift-adp` on the **Primary managed cluster** and the **Secondary managed cluster**. The DR secrets that were created when the first DRPolicy was created are similar to the following secrets. The DR secret name is preceded with the letter `v`.

```
oc get secrets -n openshift-adp
```

```
NAME                                       TYPE     DATA   AGE
v60f2ea6069e168346d5ad0e0b5faa59bb74946f   Opaque   1      3d20h
vcc237eba032ad5c422fb939684eb633822d7900   Opaque   1      3d20h
[...]
```

Note: There is one DR created secret for each managed cluster in the `openshift-adp` namespace.

5. Verify if the Data Protection Application (DPA) is already installed on each managed cluster in the OADP namespace `openshift-adp`. If not already created then follow the next step to create this resource.
   a. Create the DPA by copying the following YAML definition content to `dpa.yaml`.

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  labels:
    app.kubernetes.io/component: velero
  name: velero
  namespace: openshift-adp
spec:
  backupImages: false
  configuration:
    nodeAgent:
      enable: false
      uploaderType: restic
    velero:
      defaultPlugins:
        - openshift
        - aws
        - kubevirt
      noDefaultBackupLocation: true
```

   b. Create the DPA resource.

```
oc create -f dpa.yaml -n openshift-adp
```

```
dataprotectionapplication.oadp.openshift.io/velero created
```

   c. Verify that the OADP resources are created and are in `Running` state.

```
oc get pods,dpa -n openshift-adp
```

```
NAME                                                READY   STATUS    RESTARTS   AGE
pod/openshift-adp-controller-manager-7b64b74fcd-msjbs   1/1     Running   0          5m30s
pod/velero-694b5b8f5c-b4kwg                         1/1     Running   0          3m31s


NAME                                                 AGE
dataprotectionapplication.oadp.openshift.io/velero   3m31s
```

Creating a sample discovered application

To test failover from the **Primary managed cluster** to the **Secondary managed cluster** and relocate for discovered applications, you need a sample application that is installed without using the RHACM create application capability.

Before you begin

Note: Regional-DR now supports leveraging CephRBD volumes using non-default replica-2 storage classes that are managed by Fusion Data Foundation. Support has also been extended to compression-enabled storage classes.

Procedure

1. Log in to the **Primary managed cluster** and clone the sample application repository.

```
git clone https://github.com/red-hat-storage/ocm-ramen-samples.git
```

2. Verify that you are on the main branch.

```
cd ~/ocm-ramen-samples
git branch
* main
```

The correct directory should be used when creating the sample application based on your scenario, metro or regional. To find your directory:

```
ls workloads/deployment | egrep -v 'cephfs|k8s|base'
```

```
odr-metro-rbd
odr-regional-rbd
odr-regional-cephfs
```

3. Create a project named `busybox-discovered` on both the **Primary and Secondary managed clusters**.

```
oc new-project busybox-discovered
```

4. Create the `busybox` application on the **Primary managed cluster**.
   This sample application example is for Regional-DR using a block (Ceph RBD) volume. CephFS volumes can be used as well.

```
oc apply -k workloads/deployment/odr-regional-rbd -n busybox-discovered
```

```
persistentvolumeclaim/busybox-pvc created
deployment.apps/busybox created
```

   Note:
   Fusion Data Foundation disaster recovery solution now extends protection to discovered applications that span across multiple namespaces.

5. Verify that busybox is running in the correct project on the **Primary managed cluster**.

```
oc get pods,pvc,deployment -n busybox-discovered
```

```
NAME                              READY   STATUS    RESTARTS   AGE
pod/busybox-796fccbb95-qmxjf   1/1     Running   0          18s


NAME                                  STATUS   VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS                  VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/busybox-pvc   Bound    pvc-b20e4129-902d-47c7-b962-040ad64130c4   1Gi        RWO            ocs-
storagecluster-ceph-rbd   <unset>                 18s


NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/busybox   1/1     1            1           18
```

Enrolling a sample discovered application for disaster recovery protection

This section guides you on how to apply an existing DR Policy to a discovered application from the Protected applications tab.

Before you begin

- Ensure that Disaster Recovery has been configured and that at least one DR Policy has been created.

Procedure

1. On RHACM console, navigate to Disaster recovery > Protected applications tab.
2. Click Enroll application to start configuring existing applications for DR protection.
3. Select ACM discovered applications.
4. In the Namespace page, choose the DR cluster which is the name of the **Primary managed cluster** where `busybox` is installed.
5. Select namespace where the application is installed. For example, `busybox-discovered`.
   Note: If you have workload spread across multiple namespaces then you can select all of those namespaces to DR protect.
6. Choose a unique Name, for example `busybox-rbd`, for the discovered application and click Next.
7. In the Configuration page, select either Resource label or Recipe.
8. Resource label is used to protect your resources where you can set which resources are included in the kubernetes-object backup and what volume's persistent data is replicated. If you selected Resource label, provide label expressions and PVC label selector. Choose the label `appname=busybox` for both the kubernetes-objects and for the PVC(s).
9. If you selected Recipe, then from the Recipe list select the name of the recipe.
   Important: The recipe resource must be created in the application namespace on both managed clusters before enrolling an application for disaster recovery.
10. If you are using a recipe and want to add the recipe's parameters, under Recipe parameters, add the Key and Value (optional), then click Add parameter.
11. Click Next.
12. In the Replication page, select an existing DR Policy and the kubernetes-objects backup interval.
    Note: It is recommended to choose the same duration for the PVC data replication and kubernetes-object backup interval (for example, 5 minutes).
13. Click Next.
14. Review the configuration and click Save.
    Use the Back button to go back to the screen to correct any issues.
15. Verify that you can view the assigned policy details on the Applications page.
    a. On the Applications page, navigate to the DR Status column and view the status.
       The status can be any of the following:

       - Protecting: Shown during initial DR setup before data synchronization begins
       - Healthy: All volumes and Kubernetes resources are syncing normally
       - Warning: Volumes or Kubernetes resources are syncing slower than usual
       - Critical: Volumes or Kubernetes resources are not syncing
       - Protection Error: An error occurred during application protection setup

       You can click the status to view the last synced time for application volumes and the DR policy assigned.

       Note: It may take a few minutes for the DR Status to move from Protecting to Healthy after disaster recovery is enabled. The DR status shows Protecting during initial setup, then automatically transition to health-based statuses (Healthy/Warning/Critical) once data synchronization begins.
    b. Failover and relocate status can be viewed in the DR Status column when initiated. These statuses can be clicked to view more details about the action. DR Status shows either Protecting, Healthy, Warning, or Critical when failover or relocate is not taking place. During failover or relocate operations, the status shows the operation-specific status (FailingOver, Relocating, and so on).
16. Verify that the volume consistency groups have been enabled from the Protected applications tab:
    a. Click the arrow next to the DRPC to expand the details.
    b. Click the number under the Volume Consistency groups column to open the modal.

c. In the Volume consistency groups modal, verify the volume consistency group(s) has the correct volumes and a *Synced* status.

17. View the status of DRPC and VolumeReplicationGroup (VRG).
    a. To see the status of the DRPC, run the following command on the Hub cluster:

    ```
    oc get drpc {drpc_name} -o wide -n openshift-dr-ops
    ```

    The discovered applications store resources such as DRPlacementControl (DRPC) and Placement on the Hub cluster in a new namespace called **openshift-dr-ops**. The DRPC name can be identified by the unique Name configured in prior steps (for example, **busybox-rbd**).

    b. To see the status of the VRG for discovered applications, run the following command on the managed cluster where the busybox application was manually installed.

    ```
    oc get vrg {vrg_name} -n openshift-dr-ops
    ```

    The VRG resource is stored in the namespace **openshift-dr-ops** after a DR Policy is assigned to the discovered application. The VRG name can be identified by the unique Name configured in prior steps (for example, **busybox-rbd**).

18. Optional: For Rados block device (RBD), verify `volumereplication` and `volumegroupreplication` on the primary cluster.
    For `volumereplication`:

    ```
    oc get volumereplication -A
    ```

    Example output:

    ```
    NAMESPACE            NAME                                          AGE    VOLUMEREPLICATIONCLASS                    SOURCEKIND
    SOURCENAME                                  DESIREDSTATE   CURRENTSTATE
    busybox-discovered   vr-f9fc057c-5e95-47cb-b515-b4ee19d4ca8e   109s   rbd-volumereplicationclass-1625360775
    VolumeGroupReplication   vgr-e8b7cca8211a41455790dadc2707a08c-busybox-rbd   primary        Primary
    ```

    For `volumegroupreplication`:

    ```
    oc get volumegroupreplication -A
    ```

    Example output:

    ```
    NAMESPACE            NAME                                                VOLUMEGROUPREPLICATIONCLASS
    VOLUMEGROUPREPLICATIONCONTENT                DESIREDSTATE   CURRENTSTATE   AGE
    busybox-discovered   vgr-e8b7cca8211a41455790dadc2707a08c-busybox-rbd   rbd-volumegroupreplicationclass-1625360775-
    1602718344   vgrcontent-f9fc057c-5e95-47cb-b515-b4ee19d4ca8e   primary        Primary        113s
    ```

19. Optional: For CephFS, verify `ReplicationGroupSource` and `ReplicationSource` on the Primary cluster.
    For `ReplicationGroupSource`:

    ```
    oc get ReplicationGroupSource -A
    ```

    Example output:

    ```
    NAMESPACE        NAME                                              LAST SYNC              DURATION          NEXT SYNC
    SOURCE                                                                                                        LAST SYNC
    START
    busybox-cephfs   busybox-cephfs-bbd241b65b6589782fddaa82636e358d   2026-03-18T11:20:14Z   1m16.462467415s   2026-03-
    18T11:25:00Z   {"matchLabels":{"ramendr.openshift.io/consistency-group":"busybox-cephfs-
    bbd241b65b6589782fddaa82636e358d"}}
    ```

    For `ReplicationSource`:

    ```
    oc get ReplicationSource -A
    ```

    Example output:

    ```
    NAMESPACE        NAME          SOURCE           LAST SYNC              DURATION         NEXT SYNC
    busybox-cephfs   busybox-pvc   vs-busybox-pvc   2026-03-18T11:20:14Z   16.450725976s
    ```

20. Optional: For CephFS, verify `ReplicationGroupDestination` and `ReplicationDestination` on the Secondary cluster.
    For `ReplicationGroupDestination`:

    ```
    oc get ReplicationGroupDestination -A
    ```

    Example output:

    ```
    NAMESPACE        NAME                                              LAST SYNC              DURATION         LAST SYNC START
    busybox-cephfs   busybox-cephfs-b1f2b5bac27b3d8805437424895d212c   2026-03-18T11:20:15Z   47.700328282s   2026-03-
    18T11:20:15Z
    ```

    For `ReplicationDestination`:

    ```
    oc get ReplicationDestination -A
    ```

    Example output:

    ```
    NAMESPACE        NAME          LAST SYNC              DURATION         NEXT SYNC
    busybox-cephfs   busybox-pvc   2026-03-18T11:20:15Z   47.401909636s
    ```

Discovered application failover and relocate

Discovered application failover and relocate follows a similar process to managed applications but requires additional manual steps because RHACM does not manage the lifecycle of discovered applications.

A protected Discovered application can *Failover* or *Relocate* to its peer cluster similar to **managed applications**. However, there are some additional steps for discovered applications since RHACM does not manage the lifecycle of the application as it does for Managed applications.

This section guides you through the Failover and Relocate process for a protected discovered application.
Important: Never initiate a Failover or Relocate of an application when one or both resource types are in a Warning or Critical status.

- **Failover disaster recovery protected discovered application**
  Fail over a disaster recovery-protected discovered application to the secondary managed cluster and complete the required cleanup on the primary managed cluster.
- **Relocate disaster recovery protected discovered application**
  Relocate a disaster recovery protected discovered application from a secondary managed cluster to a primary managed cluster by using the Red Hat Advanced Cluster Management (RHACM) console.

Failover disaster recovery protected discovered application

Fail over a disaster recovery-protected discovered application to the secondary managed cluster and complete the required cleanup on the primary managed cluster.

Before you begin

- Ensure that the application namespace is created in both managed clusters (for example, `busybox-discovered`).

About this task

This procedure guides you on how to fail over a discovered application that is disaster recovery protected.

Procedure

1. In the RHACM console, navigate to Disaster Recovery > Protected applications tab.
2. At the end of the application row, click on the Actions menu and choose to initiate Failover.
3. In the Failover application modal window, review the status of the application and the target cluster.
   Important:
   If there are data inconsistencies caused by synchronization delays, a warning message appears stating **Inconsistent data on target cluster**. This alerts to the possibility of data loss if the failover is initiated. The message is no longer displayed when data synchronization is complete.
4. Click Initiate.
5. In the DR status column, you can view the progress of the failover. The modal shows the progress of the failover as Preparing > Failover > Restoring > Clean up.
6. Verify that the busybox application is running on the **Secondary managed cluster**.

   ```
   oc get pods,pvc -n busybox-discovered
   ```

   ```
   NAME                              READY   STATUS    RESTARTS   AGE
   pod/busybox-796fccbb95-qmxjf   1/1     Running   0          2m46s


   NAME                                  STATUS   VOLUME                                     CAPACITY   ACCESS MODES
   STORAGECLASS                  VOLUMEATTRIBUTESCLASS   AGE
   persistentvolumeclaim/busybox-pvc   Bound    pvc-b20e4129-902d-47c7-b962-040ad64130c4   1Gi        RWO            ocs-
   storagecluster-ceph-rbd   <unset>                 2m57s


   NAME                                                              AGE     VOLUMEREPLICATIONCLASS                 PVCNAME
   DESIREDSTATE   CURRENTSTATE
   volumereplication.replication.storage.openshift.io/busybox-pvc   2m45s   rbd-volumereplicationclass-1625360775   busybox-
   pvc   primary        Primary
   ```

7. Check the progression status of Failover until the result is `WaitOnUserToCleanup`. The DRPC name can be identified by the unique Name configured in prior steps (for example, `busybox-rbd`).

   ```
   oc get drpc {drpc_name} -n openshift-dr-ops -o jsonpath='{.status.progression}{"\n"}'
   ```

   ```
   WaitOnUserToCleanUp
   ```

   Important:
   If the result doesn't appear as `WaitOnUserToCleanup` and instead moves to `Completed`, you might encounter the error "Secondary cluster transition failed as PVC is potentially in use by a pod." This indicates that application cleanup was not completed on the primary cluster. In this case, complete the cleanup on the primary cluster to release the PVC. After cleanup, replication resumes automatically. If replication does not resume, perform a relocate operation to move the application to the secondary cluster and continue operations with the last synchronized data.

8. Remove the `busybox` application from the Primary managed cluster to complete the *Failover* process.
   a. Go to the Protected applications tab. A message to remove the application.
   b. Go to the cloned repository for `busybox` and run the following commands on the Primary managed cluster where you failed over from. Use the same directory that was used to create the application (for example, `odr-regional-rbd`).

      ```
      cd ~/ocm-ramen-samples/
      git branch
      ```

      ```
      * main
      ```

      ```
      oc delete -k workloads/deployment/odr-regional-rbd -n busybox-discovered
      ```

```
persistentvolumeclaim "busybox-pvc" deleted
deployment.apps "busybox" deleted
```

9. After deleting the application, navigate to the Protected applications tab and verify that the busybox resources are both in *Healthy* status.

Relocate disaster recovery protected discovered application

Relocate a disaster recovery protected discovered application from a secondary managed cluster to a primary managed cluster by using the Red Hat Advanced Cluster Management (RHACM) console.

Before you begin

- Ensure that the application namespace is created in both managed clusters (for example, `busybox-discovered`).

About this task

This procedure guides you on how to failover a discovered application which is disaster recovery protected.

Procedure

1. In the RHACM console, navigate to Disaster Recovery > Protected applications tab.
2. At the end of the application row, click on the Actions menu and choose to initiate Relocate.
3. In the Relocate application modal window, review the status of the application and the target cluster.
   Important:
   If there are data inconsistencies caused by synchronization delays, a warning message appears stating **Inconsistent data on target cluster**. This alerts to the possibility of data loss if the failover is initiated. The message is no longer displayed when data synchronization is complete.
4. Click Initiate.
5. In the DR status column, you can view the progress of the relocation. The progress shows Preparing > Clean up, at which time you need to perform the following clean up activities.
6. The DRPC name can be identified by the unique Name configured in prior steps (for example, `busybox-rbd`).

   ```
   oc get drpc {drpc_name} -n openshift-dr-ops -o jsonpath='{.status.progression}{"\n"}'

   WaitOnUserToCleanUp
   ```

7. Remove the busybox application from the **Secondary managed cluster** before *Relocate* to the **Primary managed cluster** is completed
   a. Go to the cloned repository for `busybox` and run the following commands on the Secondary managed cluster where you relocated from. Use the same directory that was used to create the application (for example, `odr-regional-rbd`).

   ```
   cd ~/ocm-ramen-samples/
   git branch

   * main


   oc delete -k workloads/deployment/odr-regional-rbd -n busybox-discovered

   persistentvolumeclaim "busybox-pvc" deleted
   deployment.apps "busybox" deleted
   ```

8. After deleting the application, navigate to the Protected applications tab and verify that the busybox resources are both in *Healthy* status.
9. Verify that the busybox application is running on the **Primary managed cluster**.

   ```
   oc get pods,pvc -n busybox-discovered

   NAME                              READY   STATUS    RESTARTS   AGE
   pod/busybox-796fccbb95-qmxjf   1/1     Running   0          2m46s


   NAME                                    STATUS   VOLUME                                     CAPACITY   ACCESS MODES
   STORAGECLASS                  VOLUMEATTRIBUTESCLASS   AGE
   persistentvolumeclaim/busybox-pvc   Bound    pvc-b20e4129-902d-47c7-b962-040ad64130c4   1Gi        RWO            ocs-
   storagecluster-ceph-rbd   <unset>                 2m57s


   NAME                                                          AGE     VOLUMEREPLICATIONCLASS                 PVCNAME
   DESIREDSTATE   CURRENTSTATE
   volumereplication.replication.storage.openshift.io/busybox-pvc   2m45s   rbd-volumereplicationclass-1625360775   busybox-
   pvc   primary        Primary
   ```

10. After the clean up activities are performed, the DR Status column moves from Syncing > Restoring.

Disable disaster recovery for protected applications

This section guides you to disable disaster recovery resources when you want to delete the protected applications or when the application no longer needs to be protected.

Procedure

1. Login to the Hub cluster.
2. List the `DRPlacementControl` (DRPC) resources. Each DRPC resource was created when the application was assigned a DR policy.

   ```
   oc get drpc -n openshift-dr-ops
   ```

3. Find the **DRPC** that has a name that includes the unique identifier that you chose when assigning a DR policy (for example, `busybox-rbd`) and delete the DRPC.

   ```
   oc delete {drpc_name} -n openshift-dr-ops
   ```

4. List the **Placement** resources. Each Placement resource was created when the application was assigned a DR policy.

   ```
   oc get placements -n openshift-dr-ops
   ```

5. Find the **Placement** that has a name that includes the unique identifier that you chose when assigning a DR policy (for example, `busybox-rbd-placement-1`) and delete the Placement.

   ```
   oc delete placements {placement_name} -n openshift-dr-ops
   ```

Recovering to a replacement cluster with Regional-DR

When there is a failure with the primary cluster, you get the options to either repair, wait for the recovery of the existing cluster, or replace the cluster entirely if the cluster is irredeemable. This solution guides you when replacing a failed primary cluster with a new cluster and enables failback (relocate) to this new cluster.

Before you begin

- Ensure that the Regional-DR environment has been configured with applications installed using Red Hat Advance Cluster Management (RHACM).
- Ensure that the applications are assigned a Data policy which protects them against cluster failure.

About this task

In these instructions, we are assuming that a RHACM managed cluster must be replaced after the applications have been installed and protected. For purposes of this section, the RHACM managed cluster is the **replacement cluster**, while the cluster that is not replaced is the **surviving cluster** and the new cluster is the **recovery cluster**.

Procedure

1. On the surviving cluster, backup the `clientProfileMapping`:

   ```
   oc get clientprofilemapping ocs-storagecluster -n openshift-storage -o yaml > clientProfileMappingBeforeRecovery.yaml
   ```

2. Failover all protected applications on the failed replacement cluster to the surviving cluster on the RHACM console.
   - **For managed applications**:
     You can failover protected applications using either of the following paths:

     Option 1: From the Applications page

     - On the Hub cluster, navigate to Applications. Click Actions (⋮) and select Failover application.

     Option 2: From the Protected Applications page

     - On the Hub cluster, navigate to Data Services > Disaster recovery > Protected Applications > Actions (⋮) > Failover.
   - **For discovered applications**:
     Go to All Clusters > Data Services > Disaster recovery > Protected applications tab. Click the Actions (⋮) and select Failover.
3. Validate that all protected applications are running on the surviving cluster before moving to the next step.
   Note: The PROGRESSION state for each managed application `DRPlacementControl` shows as `Cleaning Up`. For discovered applications the PROGRESSION is `WaitOnUserToCleanUp`. This is expected if the **replacement cluster** is offline or down.
4. From the Hub cluster, delete the DRCluster for the **replacement cluster**.

   ```
   oc delete drcluster <drcluster_name> --wait=false
   ```

   Note: Use `--wait=false` since the DRCluster is not deleted until a later step.
5. Remove disaster recovery for each protected application on the surviving cluster. Perform all the sub-steps on the hub cluster.
   a. For each managed application, edit the Placement and ensure that the surviving cluster is selected:

   ```
   oc edit placement <placement_name> -n namespace>
   ```

   ```
   apiVersion: cluster.open-cluster-management.io/v1beta1
   kind: Placement
   metadata:
   annotations:
     cluster.open-cluster-management.io/experimental-scheduling-disable: "true"
   [...]
   predicates:
   - requiredClusterSelector:
       claimSelector: {}
       labelSelector:
         matchExpressions:
         - key: name
           operator: In
           values:
           - cluster1  <-- Modify to be surviving cluster name
   [...]
   ```

Note: For Subscription-based applications the associated Placement can be found in the same namespace on the hub cluster similar to the managed clusters. For ApplicationSets-based applications the associated Placement can be found in the `openshift-gitops` namespace on the hub cluster.

b. For each managed application, edit the PlacementDecision and ensure that the **replacement cluster** is deleted.

```
oc edit placementdecision placementdecision_name> -n namespace> --subresource=status
```

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: PlacementDecision
metadata:
[...]
status:
  decisions:
  - clusterName: cluster2  <-- Delete this line
    reason: RetainedForFailover  <-- Delete this line
  - clusterName: cluster1
    reason: cluster1
```

Note: For Subscription-based applications the associated PlacementDecision can be found in the same namespace on the hub cluster similar to the managed clusters. For ApplicationSets-based applications the associated PlacementDecision can be found in the `openshift-gitops` namespace on the hub cluster.

c. Verify that the `s3Profile` is removed for the replacement cluster by running the following command on the surviving cluster for each protected application's VolumeReplicationGroup (VRG).
Note: For managed applications, the associated VRG is found in the application namespace. For Discovered applications the associated VRG is in the `openshift-dr-ops` namespace.

```
oc get vrg -n <vrg_namespace> -o jsonpath='{.items[0].spec.s3Profiles}' | jq
```

d. Remove disaster recovery from the applications on the RHACM console:

For managed applications:

You can remove disaster recovery using either of the following paths:

Option 1: From the Applications page

i. On the Hub cluster, navigate to All Clusters > Applications.
ii. In the Overview tab, at the end of the protected application row from the action menu (⋮), select Manage disaster recovery.
iii. Click Remove disaster recovery.
iv. Click Confirm remove.

Option 2: From the Protected Applications page

i. On the Hub cluster, navigate to All Clusters > Data Services > Disaster recovery > Protected Applications.
ii. At the end of the protected application row from the action menu (⋮), select Manage disaster recovery.
iii. Click Remove disaster recovery.
iv. Click Confirm remove.

For discovered applications:

i. Go to All Clusters > Data Services > Disaster recovery > Protected applications tab.
ii. At the end of the application row, click on the action menu (⋮) and choose Remove disaster recovery.
iii. Click Remove in the next prompt.

6. Repeat the process detailed in the last step and the sub-steps for every protected application on the surviving cluster. Removing disaster recovery for protected applications is now completed.
7. Remove all disaster recovery configurations for the **surviving cluster** and the **replacement cluster**:
   a. From the hub cluster, delete DRPolices associated with the **surviving cluster** and **replacement cluster** slated for DR removal:

```
oc delete drpolicy {drpolicy_name1 drpolicy_name2 ...}
```

   b. From the hub cluster, delete the mirrorpeer associated with the **surviving cluster** and **replacement cluster** slated for DR removal:

```
oc delete mirrorpeer {mirrorpeer_name}
```

   c. From the hub cluster, patch the mirorrpeer to remove the finalizer:

```
oc patch mirrorpeer {mirrorpeer_name} -p '{"metadata":{"finalizers":null}}' --type=merge
```

   d. From the **surviving cluster**, disable mirroring on the `cephblockpoolradosnamespace`:

```
oc patch cephblockpoolradosnamespace cephblockpoolradosnamespace_name> --type merge -p '{"spec":{"mirroring": null}}'
-n openshift-storage
```

   e. From the **surviving cluster**, disable mirroring on the `cephblockpool`:

```
oc patch cephblockpool <cephblockpool_name> --type json -p '[{"op": "replace", "path": "/spec/mirroring", "value":
{}}]' -n openshift-storage
```

8. Uninstall **Submariner** for only the **replacement cluster** (failed cluster) using the RHACM console.
   a. Go to Infrastructure > Clusters > Clustersets > Submariner add-ons view and uninstall Submariner for only the replacement cluster.
   Note: The uninstall process of Submariner for the replacement cluster (failed cluster) stays GREEN and does not complete until the replacement cluster has been detached from the RHACM console.
9. Navigate back to Clusters view and detach replacement cluster.
10. Create new OpenShift cluster (recovery cluster) and import into Infrastructure > Clusters view.
11. Add the new recovery cluster to the `Clusterset` used by Submariner.
12. Install **Submariner add-ons** only for the new **recovery cluster**.
    Note: If GlobalNet is used for the surviving cluster make sure to enable GlobalNet for the recovery cluster as well.
13. Install Fusion Data Foundation on the **recovery cluster**. The Fusion Data Foundation version should be Fusion Data Foundation 4.17 or higher and the same version of Fusion Data Foundation as on the surviving cluster.
    For managed applications using ApplicationSets, the OpenShift GitOps operator must be installed. For instructions, see Installing Red Hat OpenShift GitOps Operator in web console.

For discovered applications, the OADP operator and the Data Protection Application (DPA) must be installed. For instructions, see Prerequisites for disaster recovery protection of discovered applications.

Note: Ensure to follow the optional instructions in the documentation to modify the Fusion Data Foundation storage cluster on the recovery cluster if GlobalNet has been enabled when installing Submariner.

14. Using the RHACM console, navigate to Data Services > Disaster recovery > Policies tab.
    a. Select Create DRPolicy and name your policy.
    b. Select the **recovery cluster** and the **surviving cluster**.
    c. Create the policy. For instructions see chapter on Creating Disaster Recovery Policy on Hub cluster.
    Proceed to the next step only after the status of DRPolicy changes to `Validated`.

15. Apply the DRPolicy to the applications on the surviving cluster that were originally protected before the **replacement cluster** failed.
    Note: For discovered applications, make sure to create application namespace on the **recovery cluster** before assigning the policy.
16. Ensure that there is a `poolID` mapping between the blockpools of the **replacement cluster** and **surviving cluster**, as well as between the **failed cluster** and **recovery cluster**.
    a. Open the `clientProfileMappingBeforeRecovery.yaml` created in step 1 for editing:

    ```
    vi clientProfileMappingBeforeRecovery.yaml
    ```

    b. In the `clientProfileMappingBeforeRecovery.yaml` delete the lines and replace the values notes as follows:

    ```
    apiVersion: csi.ceph.io/v1
      kind: ClientProfileMapping
      metadata:
        creationTimestamp: "2025-11-13T07:43:45Z"  <-- Remove this line
        generation: 7  <-- Remove this line
        name: ocs-storagecluster-surviving  <-- name must be unique
        namespace: openshift-storage
        ownerReferences:  <-- Remove this line
        - apiVersion: ocs.openshift.io/v1alpha1  <-- Remove this line
          blockOwnerDeletion: true  <-- Remove this line
          controller: true  <-- Remove this line
          kind: StorageClient  <-- Remove this line
          name: ocs-storagecluster  <-- Remove this line
          uid: c9b227a4-88d7-4667-82c8-f54286f498a9  <-- Remove this line
        resourceVersion: "1392181"  <-- Remove this line
        uid: ba0676ab-d1f0-4b13-9193-db4b732280d3  <-- Remove this line
      spec:
        mappings:
        - blockPoolIdMapping:
          - - "3"  <-- Sample blockPoolId, your values may be different
            - "3"  <-- Sample blockPoolId, your values may be different
          - - "16"  <-- Sample blockPoolId, your values may be different
            - "14"  <-- Sample blockPoolId, your values may be different
          localClientProfile: openshift-storage
          remoteClientProfile: openshift-storage
    ```

    c. Save the `clientProfileMappingBeforeRecovery.yaml` file to `clientProfileMappingBeforeRecovery-surviving.yaml`.
    d. Create the `clientProfileMappingBeforeRecovery-surviving.yaml` resource on the **surviving cluster**:

    ```
    oc create -f clientProfileMappingBeforeRecovery-surviving.yaml -n openshift-storage
    ```

    e. On the **recovery cluster**, get the `blockpoolIdMapping` between the **recovery cluster** and the **surviving cluster**:

    ```
    oc get clientprofilemapping ocs-storagecluster -n openshift-storage -o yaml | yq '.spec.mappings.
    [].blockPoolIdMapping'
    ```

    Example output for two `cephblockpools`. The default is 1 `cephblockpool`. Your `blockPoolIds` may be different:

    ```
    - - "3" <-- Note this value for a future step
      - "3"
    - - "13"  <-- Note this value for a future step
      - "14"
    ```

    f. Copy the contents of the `clientProfileMappingBeforeRecovery-surviving.yaml` into a new yaml:

    ```
    cp clientProfileMappingBeforeRecovery-surviving.yaml clientProfileMappingAfterRecovery.yaml
    ```

    g. Edit the `clientProfileMappingAfterRecovery.yaml` with the correct `blockPoolIds`:

    ```
    vi clientProfileMappingAfterRecovery.yaml
    ```

    The following as an example yaml with the required changes. Replace the `blockPoolIds` with the values you noted earlier:

    ```
    apiVersion: csi.ceph.io/v1
    kind: ClientProfileMapping
    metadata:
      name: ocs-storagecluster-recovery  <-- name must be unique
      namespace: openshift-storage
    spec:
      mappings:
      - blockPoolIdMapping:
        - - "3"  <-- First index from the ClientProfileMapping on recovery cluster
          - "3"
        - - "13" <-- Second index from the ClientProfileMapping on recovery cluster
          - "14"
        localClientProfile: openshift-storage
        remoteClientProfile: openshift-storage
    ```

    h. Create the `clientProfileMappingAfterRecovery.yaml` resource on the **recovery cluster**:

    ```
    oc create -f clientProfileMappingAfterRecovery.yaml -n openshift-storage
    ```

17. Relocate the newly protected applications on the **surviving cluster** back to the new **recovery cluster** on the RHCAM console.

    For managed applications:

    You can relocate protected applications using either of the following paths:

    Option 1: From the Applications page

    a. On the Hub cluster, navigate to All Clusters > Applications.
    b. In the Overview tab, at the end of the application row from the action menu (⋮), select Manage diaster recovery.
    c. Click Relocate.

    Option 2: From the Protected Applications page

    - On the Hub cluster, navigate to Data Services > Disaster recovery > Protected Applications > Actions (⋮) > Relocate.

    For discovered applications:

    a. Go to All Clusters > Data Services > Disaster recovery > Protected applications tab.
    b. At the end of the application row, click on the action menu (⋮) and choose Relocate.

Viewing Recovery Point Objective values for disaster recovery enabled applications

Recovery Point Objective (RPO) value is the most recent sync time of persistent data from the cluster where the application is currently active to its peer. This sync time helps determine duration of data lost during failover.

About this task

You can view the Recovery Point Objective (RPO) value of all the protected volumes for their workload on the Hub cluster.
Note: This RPO value is applicable only for Regional-DR during failover. Relocation ensures there is no data loss during the operation, as all peer clusters are available.

Procedure

1. On the Hub cluster, navigate to Applications > Overview tab.
2. In the Data policy column, click the policy link for the application you applied the policy to.
   A Data Policies modal page appears with the number of disaster recovery policies applied to each application along with failover and relocation status.

3. On the Data Policies modal page, click the View more details link.
   A detailed **Data Policies** modal page is displayed that shows the policy names and the ongoing activities (Last sync, Activity status) associated with the policy that is applied to the application.

   The **Last sync time** reported in the modal page, represents the most recent sync time of all volumes that are DR protected for the application.

Hub recovery using Red Hat Advanced Cluster Management

Hub recovery using Red Hat Advanced Cluster Management enables you to use a passive hub cluster to failover or relocate disaster recovery protected workloads when the active hub is unavailable.

When your setup has active and passive Red Hat Advanced Cluster Management for Kubernetes (RHACM) hub clusters, and in case where the active hub is down, you can use the passive hub to failover or relocate the disaster recovery protected workloads.

- **Configuring passive hub cluster**
  Configure the passive hub cluster in a Regional DR environment so that disaster recovery protected workloads can be failed over or relocated when the active hub becomes unavailable.
- **Switching to passive hub cluster**
  Switch to the passive hub cluster in a Regional DR setup by restoring backups, verifying cluster imports and DRPolicy validation, and confirming workload progression when the active hub is down.

Configuring passive hub cluster

Configure the passive hub cluster in a Regional DR environment so that disaster recovery protected workloads can be failed over or relocated when the active hub becomes unavailable.

About this task

To perform hub recovery in case the active hub is down or unreachable, follow the procedure in this section to configure the passive hub cluster and then failover or relocate the disaster recovery protected workloads.

Procedure

1. Ensure that RHACM operator and MultiClusterHub is installed on the passive hub cluster. See Installation and upgrade > Install in Red Hat Advanced Cluster Management for Kubernetes product documentation.
   After the operator is successfully installed, a popover with a message that the Web console update is available appears on the user interface. Click **Refresh web console** from this popover for the console changes to reflect.

2. Before hub recovery, configure backup and restore. See Operations > Business continuity > Backup and restore in Red Hat Advanced Cluster Management for Kubernetes product documentation.
3. Install the multicluster orchestrator (MCO) operator along with Red Hat OpenShift GitOps operator on the passive RHACM hub prior to the restore. For instructions to restore your RHACM hub, see Installing Fusion Data Foundation Multicluster Orchestrator operator.
4. Ensure that `.spec.cleanupBeforeRestore` is set to `None` for the `Restore.cluster.open-cluster-management.io` resource. For details, see Operations > Business continuity > Backup and restore > Restoring a backup > Restoring passive resources while checking for backups in Red Hat Advanced Cluster Management for Kubernetes product documentation.
5. If SSL access across clusters was configured manually during setup, then re-configure SSL access across clusters. For instructions, see Configuring SSL access across clusters chapter.
6. On the passive hub, add label to `openshift-operators` namespace to enable basic monitoring of `VolumeSyncronizationDelay` alert using this command. For alert details, see Disaster recovery alerts chapter.

Switching to passive hub cluster

Switch to the passive hub cluster in a Regional DR setup by restoring backups, verifying cluster imports and DRPolicy validation, and confirming workload progression when the active hub is down.

About this task

Use this procedure when active hub is down or unreachable.

Procedure

1. During the restore procedure, to avoid eviction of resources when `ManifestWorks` are not regenerated correctly, you can enlarge the `AppliedManifestWork eviction` grace period.
   On the passive hub cluster, check for existing global `KlusterletConfig`.

   - If global **KlusterletConfig** exists then edit and set the value for `appliedManifestWorkEvictionGracePeriod` parameter to a larger value. For example, 24 hours or more.
   - If global **KlusterletConfig** does not exist, then create the **Klusterletconfig**using the following yaml:

     ```
     apiVersion: config.open-cluster-management.io/v1alpha1
     kind: KlusterletConfig
     metadata:
       name: global
     spec:
       appliedManifestWorkEvictionGracePeriod: "24h"
     ```

   The configuration is propagated to all the managed clusters automatically.

2. Restore the backups on the passive hub cluster. For information, see Operations > Business continuity > Backup and restore > Restoring a backup in Red Hat Advanced Cluster Management for Kubernetes product documentation.
   Important: Recovering a failed hub to its passive instance only restores applications and their DR protected state to its last scheduled backup. Any application that was DR protected after the last scheduled backup would need to be protected again on the new hub.
3. Verify that the restore is complete.

   ```
   oc -n <restore-namespace> wait restore <restore-name> --for=jsonpath='{.status.phase}'=Finished --timeout=120s
   ```

4. Verify that the Primary and Seconday managed clusters are successfully imported into the RHACM console and they are accessible.
   If any of the managed clusters are down or unreachable then they are not successfully imported.

   Wait until DRPolicy validation succeeds before performing any DR operation.

   If the previously down managed cluster comes back online in the future after the hub recovery process, ensure that it is successfully imported and tied to the new hub. If the cluster doesn't import and tie to the new hub, delete and recreate the `restore-acm` resource:

   ```
   oc get restore restore-acm -o yaml -n open-cluster-management-backup
   ```

   Then verify that the cluster is properly connected to the new hub.

   Note: Submariner is automatically installed once the managed clusters are imported on the passive hub.
5. Verify that the **DRPolicy** is created successfully. Run this command on the **Hub cluster** for each of the DRPolicy resources created, where *<drpolicy_name>* is replaced with a unique name.

   ```
   oc get drpolicy <drpolicy_name> -o jsonpath='{.status.conditions[].reason}{"\n"}'
   ```

   Example output:

   ```
   Succeeded
   ```

6. Refresh the RHACM console to make the DR monitoring dashboard tab accessible if it was enabled on the Active hub cluster.
7. Verify the DRPC output using the following command on the new hub cluster:

   ```
   oc get drpc -A -o wide
   ```

   If `PROGRESSION` shows a status of `PAUSED`, administrative intervention is required to unpause it. `PROGRESSION` enters `PAUSED` state under the following conditions:

   - **Cluster Query Failure**: None of the clusters were successfully queried during the DRPC reconciliation. This situation can occur during hub recovery.
   - **Action Mismatch**: The DRPC action differs from the queried VRG action.
   - **Cluster Mismatch**: The DRPC action and the VRG action are the same, but the Primary VRG is found in a different cluster than the one expected by the DRPC.

Important: If you cannot diagnose and resolve the cause of the pause, contact Red Hat Customer Support.
If `PROGRESSION` is in either `Completed` or `Cleaning up`, it is safe to proceed.

8. After all components are recovered, edit the global `KlusterletConfig` on the new hub and remove the parameter `appliedManifestWorkEvictionGracePeriod` and its value.
9. Depending on whether the active hub cluster, or both the active hub cluster along with the primary managed cluster had been down, follow the next steps based on your scenario:
    a. If only the active hub cluster had been down, and if the managed clusters are still accessible, no further action is required.
    b. If the primary managed cluster is down, along with the active hub cluster, you need to fail over the workloads from the primary managed cluster to the secondary managed cluster. For failover instructions, based on your workload type, see Subscription-based application failover between managed clusters or ApplicationSet-based application failover between managed clusters.
10. Verify that the failover is successful. When the Primary managed cluster is down, then the PROGRESSION status for the workload would be in `Cleaning Up` phase until the down managed cluster is back online and successfully imported into the RHACM console.
On the passive hub cluster, run the following command to check the PROGRESSION status.

```
oc get drpc -o wide -A

NAMESPACE               NAME                                    AGE    PREFERREDCLUSTER    FAILOVERCLUSTER     DESIREDSTATE
CURRENTSTATE    PROGRESSION    START TIME              DURATION         PEER READY
[...]
busybox                 cephfs-sub-busybox-placement-1-drpc     103m   ocp4bos1            ocp4bos2            Failover
FailedOver      Cleaning Up    2024-04-15T09:12:23Z                    False
busybox                 cephfs-sub-busybox-placement-1-drpc     102m   ocp4bos1
Deployed        Completed      2024-04-15T07:40:09Z   37.200569819s    True
[...]
```

Enabling granular disaster recovery for individual or groups of virtual machines in a namespace [Technology preview]

You can enable granular disaster recovery (DR) for individual virtual machines (VMs) or VM groups within the same namespace, allowing independent failover and relocation actions.

Important:
Granular disaster recovery for individual or groups of virtual machines in a namespace is a Technology Preview feature and is subject to Technology Preview support limitations. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information, see Technology Preview Features Support Scope.

Key highlights

- **Granular VM Control**:

  Each discovered and ACM-managed VM can have its own DR policy, enabling independent DR operations without impacting other VMs in the namespace.

- **Namespace Partitioning**:

  Supports multiple DRPCs within a namespace, avoiding the limitations of namespace-level DR protection.

- **Scenarios Supported**:
    - Independent DR for discovered and ACM-managed VMs.
    - Namespace-level DR remains for non-VM applications.
- **Upgrade Considerations**:
    - Existing DRPCs are treated as namespace-level.
    - Users must disable DR before switching to VM-specific protection.
- **Manage disaster recovery of virtual machines from Virtual Machines page**
  Note: Existing UI for namespace-level DR remains unchanged.

Important:
During failover of a CNV discovered application, if the primary managed cluster is down and other CNV applications are running in different namespaces on either of the managed clusters, there have been cases where the VM fails to start on the failover cluster. This issue is due to MAC address conflicts.

To prevent this issue, it is recommended to configure disjoint MAC address ranges for each cluster from the Day 1. For example:

```
oc patch configmap kubemacpool-mac-range-config -n openshift-cnv --type merge -p '{"data":
{"RANGE_START":"02:00:00:00:00:00","RANGE_END":"02:00:FF:ff:FF:FF"}}'

oc patch configmap kubemacpool-mac-range-config -n openshift-cnv --type merge -p '{"data":
{"RANGE_START":"02:01:00:00:00:00","RANGE_END":"02:01:FF:FF:FF:FF"}}'
```

Then restart the `kubemacpool-mac-controller-manager`.

Command example:

```
oc get pods -n openshift-cnv|grep kubemacpool-mac-controller-manager
```

Example output:

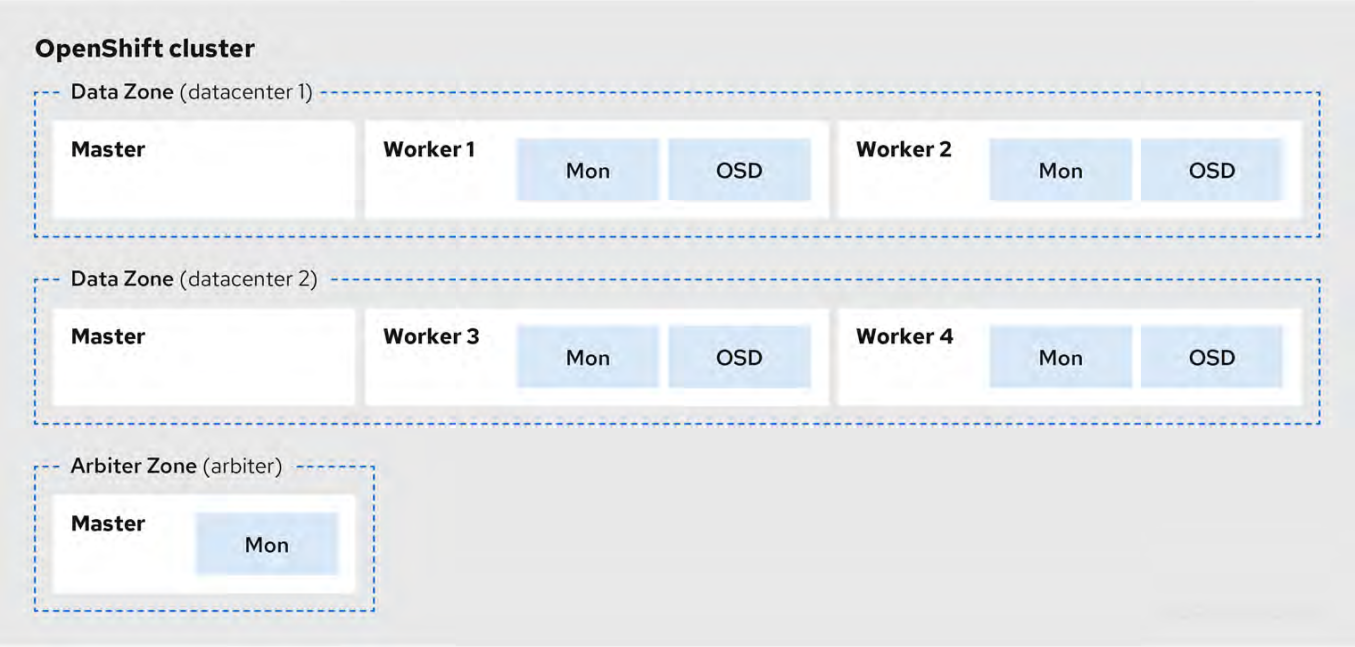
```
kubemacpool-mac-controller-manager-544ff96b4f-dgb7x    2/2     Running   3              30h
```

Disaster recovery with stretch cluster for Fusion Data Foundation

This section of the guide provides you with insights into the IBM Fusion Data Foundation Disaster Recovery (DR) solution along with necessary configuration and recovery steps for stretch clusters.

Fusion Data Foundation deployment can be stretched between two different geographical locations to provide the storage infrastructure with disaster recovery capabilities. When faced with a disaster, such as one of the two locations is either partially or fully not available, Fusion Data Foundation deployed on the OpenShift Container Platform deployment must be able to survive. This solution is available only for metropolitan spanned data centers with specific latency requirements between the servers of the infrastructure.
Note: The stretch cluster solution is designed for deployments where latencies do not exceed 10 ms maximum round-trip time (RTT) between the zones containing data volumes. For Arbiter nodes follow the latency requirements specified for etcd, see Guidance for Red Hat OpenShift Container Platform Clusters - Deployments Spanning Multiple Sites (Data Centers/Regions). Contact IBM Support if you are planning to deploy with higher latencies.
The following diagram shows the simplest deployment for a stretched cluster:

Red Hat OpenShift nodes and Fusion Data Foundation daemons



In the diagram, the Fusion Data Foundation monitor pod that is deployed in the Arbiter zone has a built-in tolerance for the master nodes. The diagram shows the master nodes in each Data Zone that are required for a highly available OpenShift Container Platform control plane. Also, it is important that the OpenShift Container Platform nodes in one of the zones has network connectivity with the OpenShift Container Platform nodes in the other two zones.

Important: You can now set up disaster recovery with stretch cluster for workloads based on OpenShift virtualization technology using Fusion Data Foundation. For more information, see OpenShift Virtualization in OpenShift Container Platform guide.

- **Requirements for enabling stretch cluster**
  Prerequisites for enabling the stretch cluster disaster recovery solution include multi-zone OpenShift nodes, latency requirements, and topology zone labels across two data zones and an arbiter zone.
- **Applying topology zone labels to OpenShift Container Platform nodes**
  Apply unique topology zone labels to OpenShift Container Platform nodes to identify the two data zones and arbiter zone required for stretch cluster deployment.
- **Installing Local Storage Operator**
  Install the Local Storage Operator from OperatorHub before creating IBM Fusion Data Foundation clusters on locally attached storage devices.
- **Installing IBM Fusion Data Foundation Operator**
  Install the IBM Fusion Data Foundation Operator from OperatorHub to deploy storage services across two data zones and an arbiter zone for stretch cluster disaster recovery.
- **Creating Fusion Data Foundation cluster**
  Create a Fusion Data Foundation cluster after installing the operator by using the OpenShift Web Console to configure internal mode with arbiter and zone awareness.
- **Verifying Fusion Data Foundation deployment**
  Verify that Fusion Data Foundation is deployed correctly for a stretch cluster by checking pod state, cluster health, and storage class availability.
- **Installing zone-aware sample application**
- **Recovering Fusion Data Foundation stretch cluster**
  Recovery methods for applications and storage in a stretch cluster depend on the application architecture and whether the storage type is ReadWriteOnce, ReadWriteMany, or StatefulSet-backed.

Requirements for enabling stretch cluster

Prerequisites for enabling the stretch cluster disaster recovery solution include multi-zone OpenShift nodes, latency requirements, and topology zone labels across two data zones and an arbiter zone.

- Ensure that you have addressed OpenShift Container Platform requirements for deployments that are spanning across multiple sites. For more information, see knowledgebase article on cluster deployments spanning multiple sites.
- Ensure that you have at least three OpenShift Container Platform master nodes in three different zones. One master node in each of the three zones.
- Ensure that you have at least four OpenShift Container Platform worker nodes that are evenly distributed across the two Data Zones.
- For stretch cluster on bare metal, use the SSD drive as the root drive for OpenShift Container Platform master nodes.
- Ensure that each node is pre-labeled with its zone label. For more information, see the Applying topology zone labels to OpenShift Container Platform nodes section.
- The stretch cluster solution is designed for deployments where latencies do not exceed 10 ms between zones. Contact IBM Support if you are planning to deploy with higher latencies.
  Note: Flexible scaling and Arbiter both cannot be enabled at the same time as they have conflicting scaling logic. With Flexible scaling, you can add one node at a time to your Fusion Data Foundation cluster. Whereas in an Arbiter cluster, you need to add at least one node in each of the two data zones.

Applying topology zone labels to OpenShift Container Platform nodes

Apply unique topology zone labels to OpenShift Container Platform nodes to identify the two data zones and arbiter zone required for stretch cluster deployment.

During a site outage, the zone that has the arbiter function makes use of the arbiter label. These labels are arbitrary and must be unique for the three locations.

For example, you can label the nodes as follows:

```
topology.kubernetes.io/zone=arbiter for Master0

topology.kubernetes.io/zone=datacenter1 for Master1, Worker1, Worker2

topology.kubernetes.io/zone=datacenter2 for Master2, Worker3, Worker4
```

- To apply the labels to the node:

  ```
  oc label node <NODENAME> topology.kubernetes.io/zone=<LABEL>
  ```

  `<NODENAME>`
  : Is the name of the node

  `<LABEL>`
  : Is the topology zone label

- To validate the labels that use the example labels for the three zones:

  ```
  oc get nodes -l topology.kubernetes.io/zone=<LABEL> -o name
  ```

  `<LABEL>`
  : Is the topology zone label

  Alternatively, you can run a single command to see all the nodes with its zone.

  ```
  oc get nodes -L topology.kubernetes.io/zone
  ```

  The stretch cluster topology zone labels are now applied to the appropriate OpenShift Container Platform nodes to define the three locations.

Installing Local Storage Operator

Install the Local Storage Operator from OperatorHub before creating IBM Fusion Data Foundation clusters on locally attached storage devices.

About this task

Install the Local Storage Operator from the Operator Hub before creating IBM Fusion Data Foundation clusters on local storage devices.

Procedure

1. Log in to the OpenShift Web Console.
2. Click Operators > OperatorHub.
3. Type `local storage` in the Filter by keyword box to find the Local Storage Operator from the list of operators, and click on it.
4. Set the following options on the Install Operator page:
   a. Update channel as `stable`.
   b. Installation mode as A specific namespace on the cluster.
   c. Installed Namespace as Operator recommended namespace openshift-local-storage.
   d. Update approval as Automatic.
5. Click Install.

What to do next

Verify that the Local Storage Operator shows a green tick that indicates successful installation.

Installing IBM Fusion Data Foundation Operator

Install the IBM Fusion Data Foundation Operator from OperatorHub to deploy storage services across two data zones and an arbiter zone for stretch cluster disaster recovery.

Before you begin

- Ensure that you have access to an OpenShift Container Platform cluster with a cluster-admin account and Operator installation permissions.
- You must have at least four worker nodes that are evenly distributed across two data centers in the Red Hat OpenShift Container Platform cluster.
- For more resource requirements, see Planning your deployment.

Important:

- To override the default cluster-wide node selector for Fusion Data Foundation, you can use the following command in command-line interface to specify a blank node selector for the **openshift-storage** namespace (create **openshift-storage** namespace in this case):

  ```
  oc annotate namespace openshift-storage openshift.io/node-selector=
  ```

- Taint a node as **infra** to ensure only IBM Fusion Data Foundation resources are scheduled on that node. This helps you save on subscription costs. For more information, see How to use dedicated worker nodes for Fusion Data Foundation chapter in the *Managing and Allocating Storage Resources* guide.

About this task

You can install Fusion Data Foundation Operator by using the Red Hat OpenShift Container Platform Operator Hub.

Procedure

1. Log in to the OpenShift Web Console.
2. Click Operators > OperatorHub.
3. Scroll or type **Fusion Data Foundation** into the Filter by keyword box to search for the Fusion Data Foundation Operator.
4. Click Install.
5. Set the following options on the Install Operator page:
   a. Update Channel as stable-4.15.
   b. Installation Mode as A specific namespace on the cluster.
   c. Installed Namespace as Operator recommended namespace openshift-storage. If Namespace **openshift-storage** does not exist, it is created during the operator installation.
   d. Select Approval Strategy as Automatic or Manual.
   If you select **Automatic** updates, then the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without any intervention.

   If you selected **Manual** updates, then the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to update the Operator to a newer version.
6. Ensure that the Enable option is selected for the Console plug-in.
7. Click Install.
8. Verification steps
   a. After the operator is successfully installed, a pop-up with a message, **Web console update is available** appears on the user interface. Click Refresh web console from this pop-up for the console changes to reflect.
   b. Go to Installed Operators and verify that the Fusion Data Foundation Operator shows a green tick indicating successful installation.
   c. Go to Storage and verify if the Data Foundation dashboard is available.

What to do next

Creating Fusion Data Foundation cluster

Creating Fusion Data Foundation cluster

Create a Fusion Data Foundation cluster after installing the operator by using the OpenShift Web Console to configure internal mode with arbiter and zone awareness.

Before you begin

Ensure that all the requirements that are mentioned in Requirements for enabling stretch cluster section are met.

About this task

Create a Fusion Data Foundation cluster after you install the Fusion Data Foundation operator.

Procedure

1. In the OpenShift Web Console, click Operators > Installed Operators to view all the installed operators.
   Ensure that the Project selected is **`openshift-storage`**.

2. Click the Fusion Data Foundation operator and then click Create StorageSystem.
3. In the Backing storage page, select the Create a new StorageClass using the local storage devices option.
4. Click Next.
   Important: You are prompted to install the Local Storage Operator if it is not already installed. Click Install, and follow the procedure as described in Installing Local Storage Operator.
5. In the Create local volume set page, provide the following information:
   a. Enter a name for the LocalVolumeSet and the StorageClass.
      By default, the local volume set name appears for the storage class name. You can change the storage class name.
      Remember: The storage class name must not exceed 40 characters and must not match any of the following predefined storage classes:
      - **`ocs-storagecluster-cephfs`**
      - **`ocs-storagecluster-ceph-rbd`**
      - **`ocs-storagecluster-ceph-rgw`**
      - **`openshift-storage.noobaa.io`**
      - **`ocs-storagecluster-ceph-rbd-virtualization`**

   b. Choose one of the following:
      - Disks on all nodes
        Uses the available disks that match the selected filters on all the nodes.

      - Disks on selected nodes
        Uses the available disks that match the selected filters only on selected nodes.
        Important:
        If the nodes selected do not match the Fusion Data Foundation cluster requirement of an aggregated 30 CPUs and 72 GiB of RAM, a minimal cluster is deployed.

        For minimum starting node requirements, see the Resource requirements section in the *Planning* guide.

   c. Select **`SSD`** or **`NVMe`** to build a supported configuration. You can select **`HDDs`** for unsupported test installations.
   d. Expand the Advanced section and set the following options:

Volume Mode	Block is selected by default.
Device Type	Select one or more device type from the dropdown list.
Disk Size	Set a minimum size of 100 GB for the device and maximum available size of the device that needs to be included.
Maximum Disks Limit	This indicates the maximum number of PVs that can be created on a node. If this field is left empty, then PVs are created for all the available disks on the matching nodes.

   e. Click Next.
      A message window to confirm the creation of LocalVolumeSet is displayed.
   f. Click Yes to continue.
6. In the **Capacity and nodes** page, configure the following:
   a. Available raw capacity is populated with the capacity value based on all the attached disks associated with the storage class. This takes some time to show up.The Selected nodes list shows the nodes based on the storage class.
   b. Select Enable arbiter checkbox if you want to use the stretch clusters. This option is available only when all the prerequisites for arbiter are fulfilled and the selected nodes are populated. For more information, see Arbiter stretch cluster requirements in Requirements for enabling stretch cluster.
   c. Select the arbiter zone from the dropdown list.
   d. Choose a performance profile for Configure performance.
      You can also configure the performance profile after the deployment using the Configure performance option from the options menu of the StorageSystems tab.
      Note: Before selecting a resource profile, make sure to check the current availability of resources within the cluster. Opting for a higher resource profile in a cluster with insufficient resources might lead to installation failures. For more information about resource requirements, see Resource requirement for performance profiles.
   e. Click Next.
7. Optional: In the Security and network page, configure the following based on your requirement:
   a. To enable encryption, select Enable data encryption for block and file storage.
   b. Select one of the following Encryption level:
      - Cluster-wide encryption to encrypt the entire cluster (block and file).
      - StorageClass encryption to create encrypted persistent volume (block only) using encryption enabled storage class.

   c. Select Connect to an external key management service checkbox. This is optional for cluster-wide encryption.
      i. From the Key Management Service Provider drop-down list, either select Vault or Thales CipherTrust Manager (using KMIP). If you selected Vault, go to the next step. If you selected Thales CipherTrust Manager (using KMIP), go to step iii.
      ii. Select an Authentication Method.
         - Using Token authentication method
         - Using Kubernetes authentication method

      iii. Enter unique Connection Name, host Address of Vault server ('https://*<hostname or ip>*'), Port number, and Token.
      iv. Expand Advanced Settings to enter additional settings and certificate details based on your **`Vault`** configuration.
         - Enter the Key Value secret path in the Backend Path that is dedicated and unique to OpenShift Data Foundation.
         - Optional: Enter TLS Server Name and Vault Enterprise Namespace.
         - Upload the respective PEM encoded certificate file to provide the CA Certificate, Client Certificate and Client Private Key .
         - Click Save and skip to step e.

   d. To use Thales CipherTrust Manager (using KMIP) as the KMS provider, complete the following steps:
      i. Enter a unique **Connection Name** for the Key Management service within the project.
      ii. In the **Address** and **Port** sections, enter the IP of Thales CipherTrust Manager and the port where the KMIP interface is enabled. For example:

         - **Address**: 123.34.3.2
         - **Port**: 5696

      iii. Upload the CA Certificate, Client Certificate and Client Private Key.

      iv. If StorageClass encryption is enabled, enter the Unique Identifier to be used for encryption and decryption generated in the previous step.

v. The TLS Server field is optional and used when there is no DNS entry for the KMIP endpoint. For example, `kmip_all_<port>.ciphertrustmanager.local`.

e. Network is set to `Default (OVN)` if you are using a single network. You can switch to Custom (Multus) if you are using multiple network interfaces and then choose any one of the following:

i. Select a Public Network Interface from the dropdown.

ii. Select a Cluster Network Interface from the dropdown.

Note: If you are using only one additional network interface, select the single `NetworkAttachementDefinition`, that is,`ocs-public-cluster` for the Public Network Interface, and leave the Cluster Network Interface blank.

f. Click Next.

8. In the Data Protection page, click Next.
9. In the Review and create page, review the configuration details.
   To modify any configuration settings, click Back to go back to the previous configuration page.
10. Click Create StorageSystem.
11. **Verification steps**
    - Verify the final Status of the installed storage cluster:
      - In the OpenShift Web Console, navigate to Installed Operators > Storage System > ocs-storagecluster-storagesystem > Resources.
      - Verify that the `Status` of `StorageCluster` is `Ready` and has a green tick mark next to it.
    - For arbiter mode of deployment:
      - In the OpenShift Web Console, navigate to Installed Operators > Storage System > ocs-storagecluster-storagesystem > Resources > ocs-storagecluster
      - In the YAML tab, search for the `arbiter` key in the `spec` section and ensure `enable` is set to `true`.

```
spec:
    arbiter:
      enable: true
    [..]
    nodeTopologies:
      arbiterLocation: arbiter #arbiter zone
    storageDeviceSets:
    - config: {}
      count: 1
        [..]
      replica: 4
status:
    conditions:
    [..]
    failureDomain: zone
```

    - To verify that all the components for Fusion Data Foundation are successfully installed, see Verifying Fusion Data Foundation deployment.

What to do next

Installing zone-aware sample application

Verifying Fusion Data Foundation deployment

Verify that Fusion Data Foundation is deployed correctly for a stretch cluster by checking pod state, cluster health, and storage class availability.

To ensure that Fusion Data Foundation is deployed correctly:

- **Verifying the state of the pods**
  Verify that all pods in the `openshift-storage` namespace are in the `Running` or `Completed` state after deploying the stretch cluster.
- **Verifying the Fusion Data Foundation cluster is healthy**
  Verify the Fusion Data Foundation cluster health by checking the Status card in the OpenShift Web Console and confirming that the Storage Cluster shows a green indicator.
- **Verifying the Multicloud Object Gateway is healthy**
  Follow this procedure to verify that the Multi cloud Object Gateway is healthy.
- **Verifying that the specific storage classes exist**
  Use this procedure to verify that storage classes are created with the Fusion Data Foundation cluster creation.

Verifying the state of the pods

Verify that all pods in the `openshift-storage` namespace are in the `Running` or `Completed` state after deploying the stretch cluster.

Procedure

1. Click Workloads> Pods from the Red Hat OpenShift Web Console.
2. Select `openshift-storage` from the Project drop-down list.
   Note: If the Show default projects option is unavailable, use the toggle button to list all the default projects.
   For more information about the expected number of pods for each component and how it varies depending on the number of nodes, see Table 1.
3. Click the Running and Completed tabs to verify that the following pods are in `Running` and `Completed` state:

Table 1. Pods corresponding to Fusion Data Foundation cluster

Component	Corresponding pods

Component	Corresponding pods
Fusion Data Foundation Operator	• **ocs-operator-*** (1 pod on any worker node) • **ocs-metrics-exporter-*** (1 pod on any worker node) • **odf-operator-controller-manager-*** (1 pod on any worker node) • **odf-console-*** (1 pod on any worker node) • **csi-addons-controller-manager-*** (1 pod on any worker node)
Rook-ceph Operator	**rook-ceph-operator-*** (1 pod on any worker node)
Multicloud Object Gateway	• **noobaa-operator-*** (1 pod on any worker node) • **noobaa-core-*** (1 pod on any storage node) • **noobaa-db-pg-cluster-1** and **noobaa-db-pg-cluster-2** (2 instances of MCG DB pod on any storage node) • **noobaa-endpoint-*** (1 pod on any storage node) • **cnpg-controller-manager-*** (1 pod on any storage node)
MON	**rook-ceph-mon-*** (5 pods are distributed across 3 zones, 2 per data-center zones and 1 in arbiter zone)
MGR	**rook-ceph-mgr-*** (2 pods on any storage node)
MDS	**rook-ceph-mds-ocs-storagecluster-cephfilesystem-*** (2 pods are distributed across 2 data-center zones)
RGW	**rook-ceph-rgw-ocs-storagecluster-cephobjectstore-*** (2 pods are distributed across 2 data-center zones)

CSI	• **cephfs**
	  • **openshift-storage.cephfs.csi.ceph.com-ctrlplugin-*** (1 pod on each worker node)
	  • **openshift-storage.cephfs.csi.ceph.com-nodeplugin-*** (2 pods distributed across worker nodes)
	• **rbd**
	  • **openshift-storage.rbd.csi.ceph.com-ctrlplugin-*** (1 pod on each worker node)
	  • **openshift-storage.rbd.csi.ceph.com-nodeplugin-*** (2 pods distributed across worker nodes)

Rook-ceph-crashcollector	**rook-ceph-crashcollector-*** (1 pod on each storage node and 1 pod in arbiter zone)
OSD	• **rook-ceph-osd-*** (1 pod for each device) • **rook-ceph-osd-prepare-ocs-deviceset-*** (1 pod for each device)

Verifying the Fusion Data Foundation cluster is healthy

Verify the Fusion Data Foundation cluster health by checking the Status card in the OpenShift Web Console and confirming that the Storage Cluster shows a green indicator.

Procedure

1. In the OpenShift Web Console, click Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the notification window that appears.
3. In the Status card of the Block and File tab, verify that *Storage Cluster* has a green tick.
4. In the Details card, verify that the cluster information is displayed.

What to do next

See Storage health levels and cluster state for information about the different health states and the alerts that appear.

Verifying the Multicloud Object Gateway is healthy

Follow this procedure to verify that the Multi cloud Object Gateway is healthy.

Procedure

1. In the OpenShift Web Console, click Storage > Data Foundation.
2. In the Status card of the Overview tab, click Storage System and then click the storage system link from the notification window that appears.
   a. In the Status card of the Object tab, verify that both *Object Service* and *Data Resiliency* have a green tick.

b. In the Details card, verify that the MCG information is displayed.

What to do next

See Storage health levels and cluster state for information about the different health states and the alerts that appear.

Important: To avoid data loss, it is recommended to take a backup of NooBaa DB PVC regularly. If NooBaa DB fails and cannot be recovered, then you can revert to the latest backed-up version. For instructions on backing up your NooBaa DB, follow the steps in the knowledgebase article, Perform a One-Time Backup of the Database for the Multicloud Object Gateway.

Verifying that the specific storage classes exist

Use this procedure to verify that storage classes are created with the Fusion Data Foundation cluster creation.

Procedure

1. Click Storage > Storage Classes from the left pane of the Red Hat OpenShift Web Console.
2. Verify that the following storage classes are created with the Fusion Data Foundation cluster creation:
   - `ocs-storagecluster-ceph-rbd`
   - `ocs-storagecluster-cephfs`
   - `openshift-storage.noobaa.io`
   - `ocs-storagecluster-ceph-rgw`

Installing zone-aware sample application

About this task

Deploy a zone-aware sample application to validate whether the Fusion Data Foundation, stretch cluster setup is configured correctly.

Important: With latency between the data zones, one can expect to see performance degradation compared to an OpenShift® Container Platform cluster with low latency between nodes and zones (for example, all nodes in the same location). How much the performance gets degraded, depends on the latency between the zones and on the application behavior that uses the storage (such as heavy write traffic). Ensure that you test the critical applications with stretch cluster configuration to ensure sufficient application performance for the required service levels.
A ReadWriteMany (rwx) Persistent Volume Claim (PVC) is created by using the `ocs-storagecluster-cephfs` storage class. Multiple pods use the newly created rwx PVC at the same time. The application that is used in the example here is called File Uploader.

This topic demonstrates how an application is spread across topology zones so that it is still available when there is a site outage.
Note: This demonstration is possible since this application shares the same rwx volume for storing files. It works for persistent data access as well because Fusion Data Foundation is configured as a stretched cluster with zone awareness and high availability.

Procedure

1. Create a new project.

   ```
   oc new-project my-shared-storage
   ```

   Note: Enure that there is an image registry available before creating the PHP application `file-uploader`.
2. Deploy the example PHP application called `file-uploader`.

   ```
   oc new-app openshift/php:7.4-ubi8~https://github.com/red-hat-storage/ocm-ramen-samples --name=file-uploader
   ```

   Example Output:

   ```
   Found image 4f2dcc0 (9 days old) in image stream "openshift/php" under tag "7.4-ubi8" for "openshift/php:7.4-
   ubi8"

   Apache 2.4 with PHP 7.4
   -----------------------
   PHP 7.4 available as container is a base platform for building and running various PHP 7.4 applications and frameworks.
   PHP is an HTML-embedded scripting language. PHP attempts to make it easy for developers to write dynamically generated web
   pages. PHP also offers built-in database integration for several commercial and non-commercial database management
   systems, so writing a database-enabled webpage with PHP is fairly simple. The most common
   use of PHP coding is probably as a replacement for CGI scripts.

   Tags: builder, php, php74, php-74

   * A source build using source code from https://github.com/christianh814/openshift-php-upload-demo will be cr
   eated
   * The resulting image will be pushed to image stream tag "file-uploader:latest"
   * Use 'oc start-build' to trigger a new build

   --> Creating resources ...
       imagestream.image.openshift.io "file-uploader" created
       buildconfig.build.openshift.io "file-uploader" created
       deployment.apps "file-uploader" created
       service "file-uploader" created
   --> Success
       Build scheduled, use 'oc logs -f buildconfig/file-uploader' to track its progress.
   ```

```
    Application is not exposed. You can expose services to the outside world by executing one or more of the commands
below:
     'oc expose service/file-uploader'

    Run 'oc status' to view your app.
```

3. View the build log and wait until the application is deployed.

   ```
   oc logs -f bc/file-uploader -n my-shared-storage
   ```

   Example Output:

   ```
   Cloning "https://github.com/christianh814/openshift-php-upload-demo" ...

       [...]
       Generating dockerfile with builder image image-registry.openshift-image-regis
   try.svc:5000/openshift/php@sha256:d97466f33999951739a76bce922ab17088885db610c
   0e05b593844b41d5494ea
   STEP 1: FROM image-registry.openshift-image-registry.svc:5000/openshift/php@s
   ha256:d97466f33999951739a76bce922ab17088885db610c0e05b593844b41d5494ea
   STEP 2: LABEL "io.openshift.build.commit.author"="Christian Hernandez <christ
   ian.hernandez@yahoo.com>"       "io.openshift.build.commit.date"="Sun Oct 1 1
   7:15:09 2017 -0700"       "io.openshift.build.commit.id"="288eda3dff43b02f7f7
   b6b6b6f93396ffdf34cb2"       "io.openshift.build.commit.ref"="master"       "
   io.openshift.build.commit.message"="trying to modularize"       "io.openshift
   .build.source-location"="https://github.com/christianh814/openshift-php-uploa
   d-demo"       "io.openshift.build.image"="image-registry.openshift-image-regi
   stry.svc:5000/openshift/php@sha256:d97466f33999951739a76bce922ab17088885db610
   c0e05b593844b41d5494ea"
   STEP 3: ENV OPENSHIFT_BUILD_NAME="file-uploader-1"     OPENSHIFT_BUILD_NAMESP
   ACE="my-shared-storage"     OPENSHIFT_BUILD_SOURCE="https://github.com/christ
   ianh814/openshift-php-upload-demo"     OPENSHIFT_BUILD_COMMIT="288eda3dff43b0
   2f7f7b6b6b6f93396ffdf34cb2"
   STEP 4: USER root
   STEP 5: COPY upload/src /tmp/src
   STEP 6: RUN chown -R 1001:0 /tmp/src
   STEP 7: USER 1001
   STEP 8: RUN /usr/libexec/s2i/assemble
   ---> Installing application source...
   => sourcing 20-copy-config.sh ...
   ---> 17:24:39     Processing additional arbitrary httpd configuration provide
   d by s2i ...
   => sourcing 00-documentroot.conf ...
   => sourcing 50-mpm-tuning.conf ...
   => sourcing 40-ssl-certs.sh ...
   STEP 9: CMD /usr/libexec/s2i/run
   STEP 10: COMMIT temp.builder.openshift.io/my-shared-storage/file-uploader-1:3
   b83e447
   Getting image source signatures

   [...]
   ```

   The command prompt returns out of the tail mode once you see `Push successful`.
   Note: The new-app command deploys the application directly from the Git repository and does not use the Red Hat OpenShift template, hence Red Hat OpenShift route resource is not created by default. You need to create the route manually.

What to do next

- Scaling the application after installation
- **Scaling the application after installation**
  Scale the application to four replicas and attach a ReadWriteMany PVC to make the application zone-aware and available across both data zones.
- **Modifying deployment to be zone-aware**
  Modify an application deployment to use topology spread constraints so that pods are distributed across availability zones and remain available during a single zone failure.

Scaling the application after installation

Scale the application to four replicas and attach a ReadWriteMany PVC to make the application zone-aware and available across both data zones.

Procedure

1. Scale the application to four replicas and expose its services to make the application zone-aware and available.

   ```
   oc expose svc/file-uploader -n my-shared-storage
   ```

   ```
   oc scale --replicas=4 deploy/file-uploader -n my-shared-storage
   ```

   ```
   oc get pods -o wide -n my-shared-storage
   ```

   You should have four file-uploader pods in a few minutes. Repeat the previous command until there are 4 file-uploader pods in the `Running` status.

2. Create a PVC and attach it into an application.

```
oc set volume deploy/file-uploader --add --name=my-shared-storage \
-t pvc --claim-mode=ReadWriteMany --claim-size=10Gi \
--claim-name=my-shared-storage --claim-class=ocs-storagecluster-cephfs \
--mount-path=/opt/app-root/src/uploaded \
-n my-shared-storage
```

This command:

- Creates a PVC.
- Updates the application deployment to include a volume definition.
- Updates the application deployment to attach a volume mount into the specified mount-path.
- Creates a new deployment with the four application pods.

3. Check the result of adding the volume.

```
oc get pvc -n my-shared-storage
```

Example Output:

```
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
AGE
my-shared-storage   Bound    pvc-5402cc8a-e874-4d7e-af76-1eb05bd2e7c7   10Gi       RWX            ocs-storagecluster-
cephfs   52s
```

Notice the **ACCESS MODE** is set to rwx.

All the four **file-uploader** pods are using the same rwx volume. Without this access mode, Red Hat OpenShift does not attempt to attach multiple pods to the same Persistent Volume (PV) reliably. If you attempt to scale up the deployments that are using ReadWriteAfter (RWO) PV, the pods may get colocated on the same node.

Modifying deployment to be zone-aware

Modify an application deployment to use topology spread constraints so that pods are distributed across availability zones and remain available during a single zone failure.

About this task

Currently, the **file-uploader** deployment is not zone-aware and can schedule all the pods in the same zone. In this case, when there is a site outage then the application is no longer available. For more information, see Configure > Nodes > Controlling pod placement onto nodes (scheduling) > Controlling pod placement by using pod topology spread constraints in Red Hat OpenShift Container Platform product documentation.

Procedure

1. Add the pod placement rule in the application deployment configuration to make the application zone-aware.
   a. Run the following command, and review the output:

```
oc get deployment file-uploader -o yaml -n my-shared-storage | less
```

Example Output:

```
[...]
spec:
  progressDeadlineSeconds: 600
  replicas: 4
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      deployment: file-uploader
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      annotations:
        openshift.io/generated-by: OpenShiftNewApp
      creationTimestamp: null
      labels:
        deployment: file-uploader
      spec:  # <-- Start inserted lines after herecontainers:  # <-- End inserted lines before here
        - image: image-registry.openshift-image-registry.svc:5000/my-shared-storage/file-
uploader@sha256:a458ea62f990e431ad7d5f84c89e2fa27bdebdd5e29c5418c70c56eb81f0a26b
          imagePullPolicy: IfNotPresent
          name: file-uploader
[...]
```

   b. Edit the deployment to use the topology zone labels.

```
oc edit deployment file-uploader -n my-shared-storage
```

Add add the following new lines between the **Start** and **End** (shown in the output in the previous step):

```
[...]
    spec:
      topologySpreadConstraints:
        - labelSelector:
```

```
            matchLabels:
              deployment: file-uploader
          maxSkew: 1
          topologyKey: topology.kubernetes.io/zone
          whenUnsatisfiable: DoNotSchedule
        - labelSelector:
            matchLabels:
              deployment: file-uploader
          maxSkew: 1
          topologyKey: kubernetes.io/hostname
          whenUnsatisfiable: ScheduleAnyway
      nodeSelector:
        node-role.kubernetes.io/worker: ""
      containers:
[...]
```

Example output:

```
deployment.apps/file-uploader edited
```

2. Scale down the deployment to **zero** pods and then back to **four** pods. This is needed because the deployment changed in terms of pod placement.
   Scaling down to **zero** pods

   ```
   oc scale deployment file-uploader --replicas=0 -n my-shared-storage
   ```

   Example output:

   ```
   deployment.apps/file-uploader scaled
   ```

   Scaling up to **four** pods

   ```
   oc scale deployment file-uploader --replicas=4 -n my-shared-storage
   ```

   Example output:

   ```
   deployment.apps/file-uploader scaled
   ```

3. Verify that the four pods are spread across the four nodes in datacenter1 and datacenter2 zones.

   ```
   oc get pods -o wide -n my-shared-storage | egrep '^file-uploader'| grep -v build | awk '{print $7}' | sort | uniq -c
   ```

   Example output:

   ```
   1 perf1-mz8bt-worker-d2hdm
      1 perf1-mz8bt-worker-k68rv
      1 perf1-mz8bt-worker-ntkp8
      1 perf1-mz8bt-worker-qpwsr
   ```

   Search for the zone labels used.

   ```
   oc get nodes -L topology.kubernetes.io/zone | grep datacenter | grep -v master
   ```

   Example output:

   ```
   perf1-mz8bt-worker-d2hdm   Ready    worker   35d   v1.20.0+5fbfd19   datacenter1
   perf1-mz8bt-worker-k68rv   Ready    worker   35d   v1.20.0+5fbfd19   datacenter1
   perf1-mz8bt-worker-ntkp8   Ready    worker   35d   v1.20.0+5fbfd19   datacenter2
   perf1-mz8bt-worker-qpwsr   Ready    worker   35d   v1.20.0+5fbfd19   datacenter2
   ```

4. Use the file-uploader web application by using your browser to upload new files.
   a. Find the route that is created.

      ```
      oc get route file-uploader -n my-shared-storage -o jsonpath --template="http://{.spec.host}{'\n'}"
      ```

      Example Output:

      ```
      http://file-uploader-my-shared-storage.apps.cluster-ocs4-abdf.ocs4-abdf.sandbox744.opentlc.com
      ```

   b. Point your browser to the web application by using the route in the previous step.
      The web application lists all the uploaded files and offers the ability to upload new ones and download the existing data.

   c. Select an arbitrary file from your local computer and upload it to the application.
      i. Click Choose file to select an arbitrary file.
      ii. Click Upload.
   d. Click List uploaded files to see the list of all currently uploaded files.

Note: The OpenShift® Container Platform image registry, ingress routing, and monitoring services are not zone-aware.

Recovering Fusion Data Foundation stretch cluster

Recovery methods for applications and storage in a stretch cluster depend on the application architecture and whether the storage type is ReadWriteOnce, ReadWriteMany, or StatefulSet-backed.

Given that the stretch cluster disaster recovery solution is to provide resiliency in the face of a complete or partial site outage, it is important to understand the different methods of recovery for applications and their storage.

How the application is designed determines how soon it becomes available again on the active zone.

There are different methods of recovery for applications and their storage depending on the site outage. The recovery time depends on the application architecture. The different methods of recovery are as follows:

- **Understanding zone failure**
  Zone failure occurs when all nodes in a data zone lose communication with the second zone; recovery succeeds only when communication between zones is fully disconnected.
- **Recovering zone-aware HA applications with rwx storage**
  Zone-aware HA applications using ReadWriteMany CephFS volumes experience pod termination in the failed zone and may require a node restart if pods enter CrashLoopBackOff during recovery.
- **Recovering HA applications with rwx storage**
  If all application replicas using ReadWriteMany storage are in the failed zone, the application takes 6–8 minutes to recover on the active zone after the pod eviction timeout expires.
- **Recovering applications with RWO storage**
  Applications using ReadWriteAfter storage have pods that become stuck in `Terminating` status after a zone failure and require manual intervention such as powering off the node to recover.
- **Recovering StatefulSet pods**
  StatefulSet pods behave like pods with ReadWriteAfter volumes during a zone failure and must be force-deleted with the node powered off to reschedule on the active zone.

Understanding zone failure

Zone failure occurs when all nodes in a data zone lose communication with the second zone; recovery succeeds only when communication between zones is fully disconnected.

In this section, zone failure is considered as a failure where all OpenShift® Container Platform, master and worker nodes in a zone are no longer communicating with the resources in the second data zone (for example, powered down nodes). If communication between the data zones is still partially working (intermittently up or down), the cluster, storage, and network admins should disconnect the communication path between the data zones for recovery to succeed.
Important: Before installing the sample application, power off the OpenShift Container Platform nodes (at least the nodes with Fusion Data Foundation devices) to test the failure of a data zone to validate that your file-uploader application is available, and you can upload new files.
Note: To prevent volume corruption during non-graceful node shutdowns, follow the steps in Fencing and tainting nodes to prevent volume corruption during non-graceful node shutdowns.

Recovering zone-aware HA applications with rwx storage

Zone-aware HA applications using ReadWriteMany CephFS volumes experience pod termination in the failed zone and may require a node restart if pods enter CrashLoopBackOff during recovery.

Applications that are deployed with `topologyKey: topology.kubernetes.io/zone`, have one or more replicas that are scheduled in each data zone, and are using shared storage, that is, ReadWriteMany (rwx) CephFS volume, terminate themselves in the failed zone after few minutes and new pods are rolled in and stuck in pending state until the zones are recovered.

An example of this type of application is detailed in the Installing zone-aware sample application section.
Important: During zone recovery if application pods go into CrashLoopBackOff (CLBO) state with permission denied error while mounting the CephFS volume, then restart the nodes where the pods are scheduled. Wait for some time and then check whether the pods are running again.
Note: To prevent volume corruption during non-graceful node shutdowns, follow the steps in Fencing and tainting nodes to prevent volume corruption during non-graceful node shutdowns.

Recovering HA applications with rwx storage

If all application replicas using ReadWriteMany storage are in the failed zone, the application takes 6–8 minutes to recover on the active zone after the pod eviction timeout expires.

Applications that are using `topologyKey: kubernetes.io/hostname` or no topology configuration, have no protection against all the application replicas being in the same zone.
Note: This can happen even with *podAntiAffinity* and `topologyKey: kubernetes.io/hostname` in the **Pod** spec because this anti-affinity rule is host-based and not zone-based.
If this happens and all replicas are located in the zone that fails, the application that uses ReadWriteMany (rwx) storage takes 6-8 minutes to recover on the active zone. This pause is for the OpenShift® Container Platform nodes in the failed zone to become `NotReady` (60 seconds) and then for the default pod eviction timeout to expire (300 seconds).

Note: To prevent volume corruption during non-graceful node shutdowns, follow the steps in Fencing and tainting nodes to prevent volume corruption during non-graceful node shutdowns.

Recovering applications with RWO storage

Applications using ReadWriteAfter storage have pods that become stuck in `Terminating` status after a zone failure and require manual intervention such as powering off the node to recover.

Applications that use ReadWriteAfter (RWO) storage have a known behavior that is described in this Kubernetes issue. Because of this issue, if there is a data zone failure, any application pods in that zone mounting RWO volumes (for example, `cephrbd` based volumes) are stuck with `Terminating` status after 6-8 minutes and is not re-created on the active zone without manual intervention.

Check the OpenShift® Container Platform nodes with a status of `NotReady`. There may be an issue that prevents the nodes from communicating with the Red Hat OpenShift control plane. However, the nodes may still be performing I/O operations against Persistent Volumes (PVs).

If two pods are concurrently writing to the same RWO volume, there is a risk of data corruption. Ensure that processes on the **NotReady** node are either terminated or blocked until they are terminated.

Example solutions:

- Use an out of band management system to power off a node, with confirmation, to ensure process termination.
- Withdraw a network route that is used by nodes at a failed site to communicate with storage.
  Note: Before restoring service to the failed zone or nodes, confirm that all the pods with PVs have terminated successfully.

To get the **Terminating** pods to re-create on the active zone, you can either force delete the pod or delete the finalizer on the associated PV. After one of these two actions is completed, the application pod should re-create on the active zone, and successfully mount its RWO storage.

Force deleting the pod
: Force deletions do not wait for confirmation from the kubelet that the pod has been terminated.

```
oc delete pod <PODNAME> --grace-period=0 --force --namespace <NAMESPACE>
```

<PODNAME>
: Is the name of the pod

<NAMESPACE>
: Is the project namespace

Deleting the finalizer on the associated PV
: Find the associated PV for the Persistent Volume Claim (PVC) that is mounted by the Terminating pod and delete the finalizer by using the **oc patch** command.

```
oc patch -n openshift-storage pv/<PV_NAME> -p '{"metadata":{"finalizers":[]}}' --type=merge
```

<PV_NAME>
: Is the name of the PV

  An easy way to find the associated PV is to describe the Terminating pod. If you see a multi-attach warning, it should have the PV names in the warning (for example, **pvc-0595a8d2-683f-443b-aee0-6e547f5f5a7c**).

```
oc describe pod <PODNAME> --namespace <NAMESPACE>
```

<PODNAME>
: Is the name of the pod

<NAMESPACE>
: Is the project namespace

  Example output:

```
[...]
Events:
  Type     Reason                  Age   From                     Message
  ----     ------                  ----  ----                     -------
  Normal   Scheduled               4m5s  default-scheduler        Successfully assigned openshift-storage/noobaa-db-pg-0
to perf1-mz8bt-worker-d2hdm
  Warning  FailedAttachVolume      4m5s  attachdetach-controller  Multi-Attach error for volume "pvc-0595a8d2-683f-443b-
aee0-6e547f5f5a7c" Volume is already exclusively attached to one node and can't be attached to another
```

Warning: Fusion Data Foundation does not support taints relating to non-graceful node shutdown for automated pod eviction and volume detachment operations. For information, see Non-graceful node shutdown handling. It is mandatory to ensure that the node is shutdown, or network route to the node is withdrawn, prior to force deleting pods.

Recovering StatefulSet pods

StatefulSet pods behave like pods with ReadWriteAfter volumes during a zone failure and must be force-deleted with the node powered off to reschedule on the active zone.

Pods that are part of a StatefulSet have a similar issue as pods mounting ReadWriteAfter (RWO) volumes. More information is referenced in the Kubernetes resource StatefulSet considerations.

To get the pods part of a StatefulSet to re-create on the active zone after 6-8 minutes you need to force delete the pod with the same requirements (that is, OpenShift® Container Platform node powered off or communication disconnected) as pods with RWO volumes.

Monitoring disaster recovery health

This section provides all the necessary configuration and commands for setting up the disaster recovery dashboard that help to monitor the health of your disaster recovery solution.

- **Enable monitoring for disaster recovery**
  Enable basic monitoring for your disaster recovery setup by adding a cluster-monitoring label to the **openshift-operators** namespace on the Hub cluster.
- **Enabling disaster recovery dashboard on Hub cluster**
  Enable the disaster recovery dashboard on the Hub cluster by creating a custom metrics allowlist configmap so that RHACM observability collects Fusion Data Foundation metrics from managed clusters.
- **Viewing health status of disaster recovery replication relationships**
  View the health status of disaster recovery replication relationships from the Hub cluster dashboard to monitor operator health, cluster status, and DR policy details

for protected applications.

- **Viewing health status of ApplicationSet-based and Subscription-based applications**
  Learn how to view the health status of ApplicationSet-based and Subscription-based applications to identify replication delays and troubleshoot issues.
- **Disaster recovery metrics**
  Ramen disaster recovery metrics, scraped by Prometheus from the hub cluster, provide per-application data on synchronization timestamps, duration, bytes transferred, and workload protection status.
- **Disaster recovery alerts**
  Reference list of disaster recovery alerts for Fusion Data Foundation.

Enable monitoring for disaster recovery

Enable basic monitoring for your disaster recovery setup by adding a cluster-monitoring label to the **openshift-operators** namespace on the Hub cluster.

About this task

Use this procedure to enable basic monitoring for your disaster recovery setup.

Procedure

1. On the Hub cluster, open a terminal window.
2. Add the following label to **openshift-operator** namespace.

```
oc label namespace openshift-operators openshift.io/cluster-monitoring='true'
```

Enabling disaster recovery dashboard on Hub cluster

Enable the disaster recovery dashboard on the Hub cluster by creating a custom metrics allowlist configmap so that RHACM observability collects Fusion Data Foundation metrics from managed clusters.

Before you begin

- Ensure that you have already installed the following:
  - Red Hat® OpenShift® Container Platform 4.16 or higher and have administrator privileges.
  - ODF Multicluster Orchestrator with the console plugin enabled.
  - Red Hat Advanced Cluster Management for Kubernetes (RHACM) 2.10 or higher from Operator Hub. For instructions on how to install, see Installation and upgrade > Install in Red Hat Advanced Cluster Management for Kubernetes product documentation.
- Ensure you have enabled observability on RHACM. See Operations > Observability > Observability service > Enabling the Observability service > Enabling Observability from the command line interface in Red Hat Advanced Cluster Management for Kubernetes product documentation.

About this task

This section guides you to enable the disaster recovery dashboard for advanced monitoring on the Hub cluster.

For Regional-DR, the dashboard shows monitoring status cards for operator health, cluster health, metrics, alerts and application count.

For Metro-DR, you can configure the dashboard to only monitor the ramen setup health and application count.

Procedure

1. On the Hub cluster, open a terminal window and perform the next steps.
2. Create the configmap file named **observability-metrics-custom-allowlist.yaml**.
   You can use the following YAML to list the disaster recovery metrics on Hub cluster. For details, see Operations > Observability > Observability service > Observability advanced confihuration > Adding custom metrics in Red Hat Advanced Cluster Management for Kubernetes product documentation. To know more about ramen metrics, see Disaster recovery metrics.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: observability-metrics-custom-allowlist
  namespace: open-cluster-management-observability
data:
  metrics_list.yaml: |
    names:
      - ceph_rbd_mirror_snapshot_sync_bytes
      - ceph_rbd_mirror_snapshot_snapshots
    matches:
      - __name__="csv_succeeded",exported_namespace="openshift-dr-system",name=~"odr-cluster-operator.*"
      - __name__="csv_succeeded",exported_namespace="openshift-operators",name=~"volsync.*"
```

3. In the **open-cluster-management-observability** namespace, run the following command:

```
oc apply -n open-cluster-management-observability -f observability-metrics-custom-allowlist.yaml
```

4. After **observability-metrics-custom-allowlist** yaml is created, RHACM starts collecting the listed Fusion Data Foundation metrics from all the managed clusters.

To exclude a specific managed cluster from collecting the observability data, add the following cluster label to the `clusters: observability: disabled`.

Viewing health status of disaster recovery replication relationships

View the health status of disaster recovery replication relationships from the Hub cluster dashboard to monitor operator health, cluster status, and DR policy details for protected applications.

Before you begin

Ensure that you have enabled the disaster recovery dashboard for monitoring. For instructions, see chapter Enabling disaster recovery dashboard on Hub cluster.

Procedure

1. On the Hub cluster, ensure that All Clusters option is selected.
2. Refresh the console to make the DR monitoring dashboard tab accessible.
3. Go to Data Services and click Data policies.
4. On the Overview tab, you can view the health status of the operators, clusters, and applications. Green tick indicates that the operators are running and available.
5. Click Disaster recovery tab to view a list of DR policy details and connected applications.

Viewing health status of ApplicationSet-based and Subscription-based applications

Learn how to view the health status of ApplicationSet-based and Subscription-based applications to identify replication delays and troubleshoot issues.

Procedure

1. On the Hub cluster, go to All Clusters > Applications.
2. In the DR Status column of the application, view the status:
   - Healthy: The last group sync time is less than 2X that of the sync interval.
   - Warning: The last group sync time is greater than 2X and less than 3X of the sync interval.
   - Critical: The last group sync time is greater than 3X of the sync interval.

What to do next

You can also view the application status for virtual machine (VM) based workloads:

- On the Hub cluster, go to All Clusters > Infrastructure > Virtual Machines > Status column.

Disaster recovery metrics

Ramen disaster recovery metrics, scraped by Prometheus from the hub cluster, provide per-application data on synchronization timestamps, duration, bytes transferred, and workload protection status.

These are the ramen metrics that are scrapped by prometheus.

- ramen_last_sync_timestamp_seconds
- ramen_policy_schedule_interval_seconds
- ramen_last_sync_duration_seconds
- ramen_last_sync_data_bytes
- ramen_workload_protection_status

Run these metrics from the Hub cluster where Red Hat Advanced Cluster Management for Kubernetes (RHACM operator) is installed.

Last synchronization timestamp in seconds

This is the time in seconds which gives the time of the most recent successful synchronization of all PVCs per application.

Metric name
: `ramen_last_sync_timestamp_seconds`

Metrics type
: Gauge

Labels

- `ObjType`: Type of the object, here its DPPC
- `ObjName`: Name of the object, here it is DRPC-Name
- `ObjNamespace`: DRPC namespace
- `Policyname`: Name of the DRPolicy
- `SchedulingInterval`: scheduling interval value from DRPolicy

Metric value

Value is set as Unix seconds which is obtained from `lastGroupSyncTime` from DRPC status.

Policy schedule interval in seconds

This gives the scheduling interval in seconds from DRPolicy.

Metric name
`ramen_policy_schedule_interval_seconds`

Metrics type
Gauge

Labels

- `Policyname`: Name of the DRPolicy

Metric value
Set to scheduling interval in seconds which is taken from DRPolicy.

Last synchronization duration in seconds

This represents the longest time taken to sync from the most recent successful synchronization of all PVCs per application.

Metric name
`ramen_last_sync_duration_seconds`

Metrics type
Gauge

Labels

- `obj_type`: Type of the object, here its DPPC
- `obj_name`: Name of the object, here it is DRPC-Name
- `obj_namespace`: DRPC namespace
- `scheduling_interval`: Scheduling interval value from DRPolicy

Metric value
The value is taken from `lastGroupSyncDuration` from DRPC status.

Total bytes transferred from most recent synchronization

This value represents the total bytes transferred from the most recent successful synchronization of all PVCs per application.

Metric name
`ramen_last_sync_data_bytes`

Metrics type
Gauge

Labels

- `obj_type`: Type of the object, here its DPPC
- `obj_name`: Name of the object, here it is DRPC-Name
- `obj_namespace`: DRPC namespace

Metric value
The value is taken from `lastGroupSyncBytes` from DRPC status.

Workload protection status

This value provides the application protection status per application that is DR protected.

Metric name
`ramen_workload_protection_status`

Metrics type
Gauge

Labels

- `obj_type`: Type of the object, here its DPPC
- `obj_name`: Name of the object, here it is DRPC-Name
- `obj_namespace`: DRPC namespace
- `scheduling_interval`: Scheduling interval value from DRPolicy

Metric value
The value is either a "1" or a "0", where "1" indicates application DR protection is healthy and a "0" indicates application protection degraded and potentially unprotected.

Disaster recovery alerts

Reference list of disaster recovery alerts for Fusion Data Foundation.

This section provides a list of all supported alerts associated with IBM Fusion Data Foundation within disaster recovery environment.

Recording rules

- Record: **ramen_sync_duration_seconds**

  Expression

  ```
  sum by (obj_name, obj_namespace, obj_type, job, policyname)(time() - (ramen_last_sync_timestamp_seconds > 0))
  ```

  Purpose
  The time interval between the volume group's last sync time and the time now in seconds.

- Record: **ramen_rpo_difference**

  Expression

  ```
  ramen_sync_duration_seconds{job="ramen-hub-operator-metrics-service"} / on(policyname, job) group_left() (ramen_policy_sc
  ```

  Purpose
  The difference between the expected sync delay and the actual sync delay taken by the volume replication group.

- Record: **count_persistentvolumeclaim_total**

  Expression

  ```
  count(kube_persistentvolumeclaim_info)
  ```

  Purpose
  Sum of all PVC from the managed cluster.

Alerts

- Alert: **VolumeSynchronizationDelay**

  Impact
  Critical

  Purpose
  Actual sync delay taken by the volume replication group is thrice the expected sync delay.

  YAML

  ```
  alert: VolumeSynchronizationDelay
  expr: ramen_rpo_difference >= 3
  for: 5s
  labels:
    severity: critical
  annotations:
    description: "The syncing of volumes is exceeding three times the scheduled snapshot interval, or the volumes have
  been recently protected. (DRPC: {{ $labels.obj_name }}, Namespace: {{ $labels.obj_namespace }})"
    alert_type: "DisasterRecovery"
  ```

- Alert: **VolumeSynchronizationDelay**

  Impact
  Warning

  Purpose
  Actual sync delay taken by the volume replication group is twice the expected sync delay.

  YAML

  ```
  alert: VolumeSynchronizationDelay
  expr: ramen_rpo_difference > 2 and ramen_rpo_difference < 3
  for: 5s
  labels:
    severity: warning
  annotations:
    description: "The syncing of volumes is exceeding two times the scheduled snapshot interval, or the volumes have
  been recently protected. (DRPC: {{ $labels.obj_name }}, Namespace: {{ $labels.obj_namespace }})"
    alert_type: "DisasterRecovery"
  ```

- Alert: **WorkloadUnprotected**

  Impact
  Warning

  Purpose
  Application protection status is degraded for more than 10 minutes.

  YAML

  ```
  alert: WorkloadUnprotected
  expr: ramen_workload_protection_status == 0
  for: 10m
  labels:
    severity: warning
  annotations:
    description: "Workload is not protected for disaster recovery (DRPC: {{ $labels.obj_name }}, Namespace: {{
  $labels.obj_namespace }})."
    alert_type: "DisasterRecovery"
  ```

Troubleshooting disaster recovery

This troubleshooting section provides guidance or workarounds on how to fix some of the disaster recovery configuration issues.

- **Troubleshooting Metro-DR**
  Administrators can use this troubleshooting information to understand how to troubleshoot and fix their Metro-DR solution.
- **Troubleshooting Regional-DR**
  Administrators can use this troubleshooting information to understand how to troubleshoot and fix their Regional-DR solution.
- **Troubleshooting 2-site stretch cluster with Arbiter**
  Administrators can use this troubleshooting information to understand how to troubleshoot and fix their 2-site stretch cluster with arbiter environment.

Troubleshooting Metro-DR

Administrators can use this troubleshooting information to understand how to troubleshoot and fix their Metro-DR solution.

- **A StatefulSet application stuck after failover**
  Troubleshoot a StatefulSet application stuck after failover.
- **DR policies protect all applications in the same namespace**
  Troubleshoot DR policies that protect all applications in the same namespace.
- **During failback of an application stuck in Relocating state**
  Troubleshoot an application stuck in a *Relocating* state during failback.
- **Relocate or failback might be stuck in Initiating state**
  If a relocate or failback operation is stuck in `Initiating` state after the primary cluster recovers while the secondary goes down, the ApplicationSet controller on the old active hub may need to be scaled down.

A StatefulSet application stuck after failover

Troubleshoot a StatefulSet application stuck after failover.

Problem
: While relocating to a preferred cluster, DRPlacementControl is stuck reporting PROGRESSION as *MovingToSecondary*.

  Previously, before Kubernetes v1.23, the Kubernetes control plane never cleaned up the PVCs created for StatefulSets. This activity was left to the cluster administrator or a software operator managing the StatefulSets. Due to this, the PVCs of the StatefulSets were left untouched when their Pods are deleted. This prevents Ramen from relocating an application to its preferred cluster.

Resolution

1. If the workload uses StatefulSets, and relocation is stuck with PROGRESSION as *MovingToSecondary*, then run:

   ```
   oc get pvc -n <namespace>
   ```

2. For each bounded PVC for that namespace that belongs to the StatefulSet, run:

   ```
   oc delete pvc <pvcname> -n namespace
   ```

   After all PVCs are deleted, Volume Replication Group (VRG) transitions to secondary, and then gets deleted.

3. Run the following command:

   ```
   oc get drpc -n <namespace> -o wide
   ```

   The PROGRESSION reports *Completed* and the relocation is complete.
   Note: Changing to *Completed* can take anywhere between a few seconds up to a few minutes.

Result
: The workload is relocated to the preferred cluster

BZ reference: [2118270]

DR policies protect all applications in the same namespace

Troubleshoot DR policies that protect all applications in the same namespace.

Problem
: While only single application is selected to be used by a DR policy, all applications in the same namespace are protected. This results in PVCs, that match the `DRPlacementControl` `spec.pvcSelector` across multiple workloads or if the selector is missing across all workloads, replication management to potentially manage each PVC multiple times and cause data corruption or invalid operations based on individual `DRPlacementControl` actions.

Resolution
: Label PVCs that belong to a workload uniquely, and use the selected label as the DRPlacementControl `spec.pvcSelector` to disambiguate which DRPlacementControl protects and manages which subset of PVCs within a namespace. It is not possible to specify the `spec.pvcSelector` field for the DRPlacementControl using the user interface, hence the DRPlacementControl for such applications must be deleted and created using the command line.

BZ reference: [2128860]

During failback of an application stuck in Relocating state

Troubleshoot an application stuck in a *Relocating* state during failback.

Problem
: This issue might occur after performing failover and failback of an application (all nodes or cluster are up). When performing failback application stuck in the *Relocating* state with a message of `Waiting` for PV restore to complete.

Resolution
: Use S3 client or equivalent to clean up the duplicate PV objects from the S3 store. Keep only the one that has a timestamp closer to the failover or relocate time.

BZ reference: [2120201]

Relocate or failback might be stuck in Initiating state

If a relocate or failback operation is stuck in **`Initiating`** state after the primary cluster recovers while the secondary goes down, the ApplicationSet controller on the old active hub may need to be scaled down.

Problem
: When a primary cluster is down and comes back online while the secondary goes down, **`relocate`** or **`failback`** might be stuck in the **`Initiating`** state.

Resolution
: To avoid this situation, cut off all access from the old active hub to the managed clusters.
Alternatively, you can scale down the ApplicationSet controller on the old active hub cluster either before moving workloads or when they are in the clean-up phase.
On the old active hub, scale down the two deployments using the following commands:

```
oc scale deploy -n openshift-gitops-operator openshift-gitops-operator-controller-manager --replicas=0
oc scale statefulset -n openshift-gitops openshift-gitops-application-controller --replicas=0
```

BZ reference: [2243804]

Troubleshooting Regional-DR

Administrators can use this troubleshooting information to understand how to troubleshoot and fix their Regional-DR solution.

- **rbd-mirror daemon health is in warning state**
  If the **`rbd-mirror`** daemon health reports a **`warning`** state after a network disconnection, the daemon may have a startup issue caused by unreliable connectivity to the secondary cluster.
- **volsync-rsync-src pod is in error state as it is unable to resolve the destination hostname**
  If the **`volsync-rsync-src`** pod enters an error state with an SSH hostname resolution failure, the **`submariner-lighthouse-agent`** may need to be restarted on both nodes.
- **Cleanup and data sync for ApplicationSet workloads remain stuck after older primary managed cluster is recovered post failover**
  Troubleshoot DR policies that protect all applications in the same namespace.
- **VM recipe known issues**
  Lists known VM recipe issues.

rbd-mirror daemon health is in warning state

If the **`rbd-mirror`** daemon health reports a **`warning`** state after a network disconnection, the daemon may have a startup issue caused by unreliable connectivity to the secondary cluster.

Problem

There appears to be numerous cases where WARNING gets reported if mirror service **`::get_mirror_service_status`** calls **`Ceph`** monitor to get service status for **`rbd-mirror`**.

Following a network disconnection, **`rbd-mirror`** daemon health is in the **`warning`** state while the connectivity between both the managed clusters is fine.

Resolution

Run the following command in the toolbox and look for **`leader:false`**

```
rbd mirror pool status --verbose ocs-storagecluster-cephblockpool | grep 'leader:'
```

If you see the following in the output:

Output	Workaround
leader: false	It indicates that there is a daemon startup issue and the most likely root cause could be due to problems reliably connecting to the secondary cluster. Workaround: Move the **`rbd-mirror`** pod to a different node by deleting the pod and verify that it has been rescheduled on another node.
`leader: true` or no output	Contact IBM Support

BZ reference: [2118627]

volsync-rsync-src pod is in error state as it is unable to resolve the destination hostname

If the `volsync-rsync-src` pod enters an error state with an SSH hostname resolution failure, the `submariner-lighthouse-agent` may need to be restarted on both nodes.

Problem

VolSync source pod is unable to resolve the hostname of the VolSync destination pod. The log of the VolSync Pod consistently shows an error message over an extended period of time similar to the following log snippet.

```
oc logs -n busybox-workloads-3-2 volsync-rsync-src-dd-io-pvc-1-p25rz
```

Example output

```
VolSync rsync container version: ACM-0.6.0-ce9a280
Syncing data to volsync-rsync-dst-dd-io-pvc-1.busybox-workloads-3-2.svc.clusterset.local:22 ...
ssh: Could not resolve hostname volsync-rsync-dst-dd-io-pvc-1.busybox-workloads-3-2.svc.clusterset.local: Name or service not
known
```

Resolution

Restart `submariner-lighthouse-agent` on both nodes.

```
oc delete pod -l app=submariner-lighthouse-agent -n submariner-operator
```

Cleanup and data sync for ApplicationSet workloads remain stuck after older primary managed cluster is recovered post failover

Troubleshoot DR policies that protect all applications in the same namespace.

Problem

ApplicationSet based workload deployments to managed clusters are not garbage collected in cases when the hub cluster fails. It is recovered to a standby hub cluster, while the workload has been failed over to a surviving managed cluster. The cluster that the workload was failed over from, rejoins the new recovered standby hub.

ApplicationSets that are DR protected, with a regional DRPolicy, hence starts firing the `VolumeSynchronizationDelay` alert. Further such DR protected workloads cannot be failed over to the peer cluster or relocated to the peer cluster as data is out of sync between the two clusters.

Resolution

The workaround requires that

`openshift-gitops`
operators can own the workload resources that are orphaned on the managed cluster that rejoined the hub post a failover of the workload was performed from the new recovered hub. To achieve this the following steps can be taken:

1. Determine the Placement that is in use by the ArgoCD ApplicationSet resource on the hub cluster in the `openshift-gitops` namespace.
2. Inspect the placement label value for the ApplicationSet in this field: `spec.generators.clusterDecisionResource.labelSelector.matchLabels`. This would be the name of the Placement resource *<placement-name>*
3. Ensure that there exists a `PlacemenDecision` for the ApplicationSet referenced `Placement`.

   ```
   oc get placementdecision -n openshift-gitops --selector cluster.open-cluster-management.io/placement=<placement-name>
   ```

   This results in a single `PlacementDecision` that places the workload in the currently desired failover cluster.
4. Create a new `PlacementDecision` for the ApplicationSet pointing to the cluster where it should be cleaned up.

   For example:

   ```
   apiVersion: cluster.open-cluster-management.io/v1beta1
   kind: PlacementDecision
   metadata:
     labels:
       cluster.open-cluster-management.io/decision-group-index: "1" # Typically one higher than the same value in the
   esisting PlacementDecision determined at step (2)
       cluster.open-cluster-management.io/decision-group-name: ""
       cluster.open-cluster-management.io/placement: cephfs-appset-busybox10-placement
     name: <placemen-name>-decision-<n> # <n> should be one higher than the existing PlacementDecision as determined in
   step (2)
     namespace: openshift-gitops
   ```
5. Update the newly created `PlacementDecision` with a status **subresource**.

   ```
   decision-status.yaml:
   status:
     decisions:
     - clusterName: <managedcluster-name-to-clean-up> # This would be the cluster from where the workload was failed
   over, NOT the current workload cluster
       reason: FailoverCleanup

   oc patch placementdecision -n openshift-gitops <placemen-name>-decision-<n> --patch-file=decision-status.yaml --
   subresource=status --type=merge
   ```

6. Watch and ensure that the Application resource for the ApplicationSet has been placed on the desired cluster.

   ```
   oc get application -n openshift-gitops  <applicationset-name>-<managedcluster-name-to-clean-up>
   ```

   In the output, check if the SYNC STATUS shows as **Synced** and the HEALTH STATUS shows as **Healthy**.

7. Delete the PlacementDecision that was created in step (3), such that ArgoCD can garbage collect the workload resources on the <managedcluster-name-to-clean-up>

   ```
   oc delete placementdecision -n openshift-gitops <placemen-name>-decision-<n>
   ```

ApplicationSets that are DR protected, with a regional DRPolicy, stops firing the **VolumeSynchronizationDelay** alert.

BZ reference: [2268594]

VM recipe known issues

Lists known VM recipe issues.

`ramendr.openshift.io/k8s-resource-selector` label is not automatically applied when a PVC is added to an already protected workload NS

Problem
: When an additional PV or PVC is attached to an already protected VM on Day 2, the newly created PV/PVC is not automatically protected.

Resolution
: To ensure the new PV/PVC is protected, manually add the following label on the new PV/PVC:

  ```
  ("ramendr.openshift.io/k8s-resource-selector": "<value same as that of base PVC of associated VM>")
  ```

  This label links the new PV/PVC to the existing protected group of resources.

DRPolicy without flattening applied to a cloned volume leads to errors

Problem
: Regional-DR currently does not support volumes created from CSI snapshots or clones in protected applications. These volumes rely on a parent volume that isn't replicated.

Resolution
: To avoid these issues, use a DRPolicy that flattens PVs before replication if cloning is required.

  For more information, see this knowledgebase article.

Troubleshooting 2-site stretch cluster with Arbiter

Administrators can use this troubleshooting information to understand how to troubleshoot and fix their 2-site stretch cluster with arbiter environment.

- **Recovering workload pods stuck in ContainerCreating state post zone recovery**
  If workload pods remain stuck in **ContainerCreating** state after a complete zone failure and recovery, a CSI driver error or stale mount path may be preventing the volume from being attached.

Recovering workload pods stuck in ContainerCreating state post zone recovery

If workload pods remain stuck in **ContainerCreating** state after a complete zone failure and recovery, a CSI driver error or stale mount path may be preventing the volume from being attached.

Problem

After performing complete zone failure and recovery, the workload pods are sometimes stuck in **ContainerCreating** state with the any of the following errors:

- MountDevice failed to create newCsiDriverClient: driver name openshift-storage.rbd.csi.ceph.com not found in the list of registered CSI drivers
- MountDevice failed for volume <volume_name> : rpc error: code = Aborted desc = an operation with the given Volume ID <volume_id> already exists
- MountVolume.SetUp failed for volume <volume_name> : rpc error: code = Internal desc = staging path <path> for volume <volume_id> is not a mountpoint

Resolution

If the workload pods are stuck with any of the preceding errors, perform the following workarounds:

- For **ceph-fs** workload stuck in **ContainerCreating**:
  1. Restart the nodes where the stuck pods are scheduled
  2. Delete these stuck pods
  3. Verify that the new pods are running
- For **ceph-rbd** workload stuck in **ContainerCreating** that do not self recover after sometime
  1. Restart csi-rbd plugin pods in the nodes where the stuck pods are scheduled

2. Verify that the new pods are running

Disabling disaster recovery for a disaster recovery-enabled application

Learn how to disable disaster recovery for disaster recovery-enabled managed and discovered applications.

- **Disabling disaster recovery for managed applications**
  Use this procedure to disable disaster recovery for disaster recovery-managed applications.
- **Disabling disaster recovery for discovered applications**
  Use this procedure to disable disaster recovery for disaster recovery-discovered applications.

Disabling disaster recovery for managed applications

Use this procedure to disable disaster recovery for disaster recovery-managed applications.

Procedure

1. In the RHACM console, go to All Clusters > Data Services > Protected applications tab.
2. At the end of the application row, click Acions menu and select Remove disaster recovery.
3. Click Remove.

Results

Your application loses disaster recovery protection, preventing volume synchronization (replication) between clusters.

The application is no longer in the Protected applications tab when the disaster recovery is removed.

Disabling disaster recovery for discovered applications

Use this procedure to disable disaster recovery for disaster recovery-discovered applications.

Procedure

1. In the RHACM console, go to All Clusters > Data Services > Protected applications tab.
2. At the end of the application row, click Acions menu and select Remove disaster recovery.
3. Click Remove.

Results

Your application loses disaster recovery protection, preventing volume synchronization (replication) between clusters.

The application is no longer in the Protected applications tab when the disaster recovery is removed.

Uninstalling disaster recovery from hub and managed clusters

Learn how to uninstall disaster recovery (DR) partially or completely from hub and managed clusters.

You can uninstall Regional-DR and Metro-DR in the following ways:

Partial uninstallation

Partial uninstallation involves removing all DR resources and configurations from two or more peer-managed clusters that are defined in a DRPolicy.

Complete uninstallation

Complete uninstallation is a full uninstallation of the DR setup and includes the following steps:

- Remove all DRPolicies and mirrorpeers from all peer clusters.
- Remove the DR operators and associated resources from the hub cluster.

- **Prerequisites for disaster recovery uninstallation**
  Before uninstalling disaster recovery (DR), first remove disaster recovery protection from all protected applications, both managed and discovered.
- **Uninstalling disaster recovery for peer-managed clusters**
  Learn how to uninstall disaster recovery (DR) resources for the peer-managed clusters.
- **Uninstalling disaster recovery operators**
  Learn how to uninstall disaster recovery (DR) completely from DR operators.

Prerequisites for disaster recovery uninstallation

Before uninstalling disaster recovery (DR), first remove disaster recovery protection from all protected applications, both managed and discovered.

Before you begin

Ensure that both managed clusters are online and healthy.

Procedure

1. Find all DRPolicies associated with the peer clusters you want to remove DR from using one of the following options:
   - **Using the RHACM console**:
     - On the hub cluster, go to All Clusters > Data Services > Disaster Recovery > Policies.
     - Identify any DRPolicies associated with the peer clusters you want to remove DR from.
   - **Using the CLI**:
     - Get the names of the peer clusters from the RHACM console. On the hub cluster, go to All Clusters > Infrastructure > Clusters.
     - Run the following commands on the hub cluster, replacing the cluster names with the peer clusters found in the previous step:

       ```
       CLUSTER_A="{cluster1_name}"

       CLUSTER_B="{cluster2_name}"

       oc get drpolicies -o json | jq -r --arg c1 "$CLUSTER_A" --arg c2 "$CLUSTER_B" '.items[]
       |select((.spec.drClusters | sort) == ([$c1, $c2] | sort)) .metadata.name'
       ```

     - Identify any DRPolicies associated with the peer clusters you want to remove DR from.
2. Find any existing DRPlacementControl (DRPC) resources that are associated with your peer-managed clusters.
   Run the following command on the hub cluster for each DRPolicy identified in step 1:

   ```
   DRPOLICY_NAME="{drpolicy_name}"
   ```

   Replace **drpolicy_name** with the DRPolicy.

   ```
   oc get drplacementcontrols -A -o json | jq -r --arg drpc "$DRPOLICY_NAME" \ '.items[] | select(.spec.drPolicyRef.name ==
   $drpc) | "\(.metadata.namespace)/\(.metadata.name)"'
   ```

3. If any DRPCs are found, remove DR protection from the associated applications.
   a. Remove DR protection from managed applications.
      i. On the hub cluster, go to All Clusters > Applications.
      ii. In the Overview tab, at the end of the protected application row from the action menu **(⋮)**, select Manage disaster recovery.
      iii. Click Remove disaster recovery.
      iv. Click Confirm remove.
   b. Remove DR protection from discovered applications.
      i. In the RHACM console, go to All Clusters > Data Services > Protected applications tab.
      ii. At the end of the application row, click the action menu **(⋮)** and select Remove disaster recovery.
      iii. Click Remove in the next prompt.
4. Run the following command on the hub cluster to find the associated mirrorpeer:

   ```
   export DR_POLICY={drpolicy_name}
   ```

   Replace **drpolicy_name** with one of the DRPolicy names found in step 1.

   ```
   DR_CLUSTERS=$(oc get drpolicy $DR_POLICY -o json | jq -r '.spec.drClusters | sort | join(",")')

   oc get mirrorpeers -A -o json | jq -r --arg dr_clusters "$DR_CLUSTERS" '.items[] | {name: .metadata.name,clusters:
   (.spec.items | map(.clusterName) | sort | join(",")) } |select(.clusters == $dr_clusters) | .name'
   ```

Uninstalling disaster recovery for peer-managed clusters

Learn how to uninstall disaster recovery (DR) resources for the peer-managed clusters.

Before you begin

Ensure that the cluster names, DRPolicies, and mirrorpeer are identified as instructed in Prerequisites for disaster recovery uninstallation.

Procedure

1. From the hub cluster, delete DRPolices associated with the peer-managed clusters that are slated for DR removal.
   The DRPolicies that are found in step 1 of Prerequisites for disaster recovery uninstallation needs to be deleted.

   Command example:

   ```
   oc delete drpolicy {drpolicy_name1 drpolicy_name2 ...}
   ```

   Replace **drpolicy_name1** with the DRPolicy names that are found in step 2 of Prerequisites for disaster recovery uninstallation needs to be deleted.

2. From the hub cluster, delete the mirrorpeer that is associated with the peer-managed clusters that are slated for DR removal.

One mirrorpeer per DRPolicy that are found in step 4 of Prerequisites for disaster recovery uninstallation needs to be deleted.

Command example:

```
oc delete mirrorpeer {mirrorpeer_name}
```

Important: If you are removing DR for specific peer clusters, do not proceed to uninstall DR operators or other shared resources. This concludes the partial DR uninstall process.

Uninstalling disaster recovery operators

Learn how to uninstall disaster recovery (DR) completely from DR operators.

Before you begin

If your goal is to completely remove DR from your OpenShift® environment, ensure that you followed all the steps that are mentioned in Prerequisites for disaster recovery uninstallation and Uninstalling disaster recovery for peer-managed clusters to uninstall DR from every peer-managed cluster.

Procedure

1. On the hub cluster, delete the `subscriptions.operators` and `clusterserviceversion` (csv) for the Red Hat® OpenShift Data Foundation Multicluster Orchestrator and OpenShift DR Hub Operator.
   Command example:

   ```
   oc delete subscriptions.operators.coreos.com -l operators.coreos.com/odr-hub-operator.openshift-operators -n openshift-operators
   oc delete subscriptions.operators.coreos.com -l operators.coreos.com/odf-multicluster-orchestrator.openshift-operators -n openshift-operators
   oc delete csv -l operators.coreos.com/odr-hub-operator.openshift-operators -n openshift-operators
   oc delete csv -l operators.coreos.com/odf-multicluster-orchestrator.openshift-operators -n openshift-operators
   ```

2. Alternatively, you can delete the Red Hat OpenShift Data Foundation Multicluster Orchestrator and OpenShift DR Hub Operator from the RHACM console.
   a. Go to Operators > Installed Operators.
   b. Locate the DR Hub Operator.
   c. Click the action menu (⋮) and select Uninstall Operator.
   d. Repeat the same steps for the Red Hat OpenShift Data Foundation Multicluster Orchestrator.
3. From the hub cluster, delete the `managedclusterview` for each peer cluster.
   Command example:

   ```
   oc delete managedclusterview -l multicluster.odf.openshift.io/created-by=odf-multicluster-managedcluster-controller -A
   ```

4. From the hub cluster, delete the dynamic console plug-in for the Red Hat OpenShift Data Foundation Multicluster Orchestrator operator.
   Command example:

   ```
   oc delete consoleplugins.console.openshift.io odf-multicluster-console
   ```

5. From the hub cluster, delete the configmap `odf-client-info`.
   Command example:

   ```
   oc delete configmap odf-client-info -n openshift-operators
   ```

6. From the hub cluster, delete the `odf-multicluster-console` service.
   Command example:

   ```
   oc delete service odf-multicluster-console -n openshift-operators
   ```

Results

All DR-related operator components and configurations are fully removed from the hub cluster and all managed clusters.

Multi-cluster IBM Fusion using Hosted Control Plane

You can create Hosted Control Plane clusters with OpenShift® Virtualization on IBM Fusion and deploy IBM Fusion services on those clusters.

A single IBM Fusion hub deployment on Bare Metal, Amazon Web Services (AWS), or IBM Z can support IBM Fusion services on OpenShift Hosted Control Plane clusters. Hosted Control Planes enable the nodes in the host Red Hat® OpenShift Container Platform cluster to host multiple clusters that are virtualized using OpenShift Virtualization. With this capability, you can create clusters that match specific workload requirements, with clusters created on demand.

With Hosted Control Planes for Red Hat OpenShift Container Platform, control planes are created as pods on a hosting cluster, removing the need for dedicated virtual or physical machines for each control plane.

For Hosted Control Plane deployments, clusters are created using the IBM Fusion hub cluster, with storage provided by Fusion Data Foundation. The Fusion Data Foundation service must be deployed on the host (base) cluster to enable the deployment of Hosted Control Plane clusters.

In this architecture:

- IBM Fusion is referred to as the hub cluster.
- IBM Fusion spokes are referred to as hosted clusters.

To create hosted clusters, see Hosted control plane.

To deploy a storage client in the Hosted Control Plane clusters, see Installing Fusion Data Foundation on a Hosted Control Plane cluster.

Important:

- IBM Fusion supports Bare Metal Hosted Control Plane and virtual machine based multi-cluster using KubeVirt. IBM Fusion supports Hosted Control Plane deployments on Bare Metal, Amazon Web Services (AWS), and IBM Z, including virtual machine–based multi-cluster deployments using OpenShift Virtualization (KubeVirt).
- Backup and restore of Hosted Control Plane clusters are not supported in IBM Fusion. For information about backup and restore of Hosted Control Plane clusters, see Hosted control planes section in *Red Hat OpenShift Container Platform* documentation.
- IBM Cloud Paks and IBM Maximo® applications are supported on Hosted Control Plane clusters in IBM Fusion. For more information, see IBM Cloud Paks support for IBM Fusion and IBM Maximo® Applications support for IBM Fusion.
- Bare Metal and virtualized Hosted Control Plane clusters are supported in connected environments, including environments that use a proxy.
- Hosted Control Plane deployments are supported in both connected environments (including environments that use a proxy) and offline (disconnected) environments.

- **Interoperability matrix**
  Use this section to get the detailed support matrix for the Hosted Control Plane.
- **Prerequisites for Hosted Control Plane clusters**
  The prerequisites before you create virtualized or Bare Metal Hosted Control Plane clusters.
- **Configuring a Hosted Control Plane cluster in a proxy environment**
  Use these instructions to gather the required configuration and define the necessary Kubernetes resources to configure a `HostedCluster` in a proxy environment.
- **Deployment of virtualized clusters with Fusion Data Foundation**
  Create virtualized (Hosted Control Plane) clusters with the Fusion Data Foundation as storage type.
- **Deploying Bare Metal Hosted Control Plane cluster in disconnected environment**
  A disconnected environment deployment operates without internet access. You can deploy Hosted Control Plane (HCP) in a disconnected environment on Bare Metal infrastructure.
- **Deploying virtualized Hosted Control Plane cluster in disconnected environment**
  A disconnected environment deployment operates without internet access. You can deploy Hosted Control Plane (HCP) in a disconnected environment on the virtualized infrastructure.
- **Deployment of Bare Metal clusters with Fusion Data Foundation**
  Create Bare Metal hosted clusters with Fusion Data Foundation as the storage type. It supports both Bare Metal internal servers in a rack and Bare Metal external servers.
- **Installing the IBM Fusion base**
  Procedure to install the IBM Fusion base on a Hosted Control Plane cluster.
- **Installation of data services on a Hosted Control Plane cluster**
  Provision Hosted Control Plane clusters with services like Fusion Data Foundation, Global Data Platform remote mount, and Backup & Restore.
- **Backup & Restore of applications in Hosted Control Plane clusters**
  Using the Backup & Restore service, you can backup and restore your applications on Hosted Control Plane clusters.
- **Updating the storage quota of a Hosted Control Plane cluster**
  Update the allotted storage quota of the Fusion Data Foundation client on a Hosted Control Plane cluster to meet the storage requirement of the application workloads.
- **Cleaning up of a Hosted Control Plane cluster**
  If you no longer need a hosted cluster, use this procedure to remove it.
- **Troubleshooting issues and known limitations in Hosted Control Plane cluster**
  Use these troubleshooting information to know the problem and workaround for the installation of spoke cluster and

Interoperability matrix

Use this section to get the detailed support matrix for the Hosted Control Plane.

Run the following command to retrieve the list of supported OpenShift® Container Platform versions for an Hosted Control Plane cluster with the installed MCE version.

```
oc get configmap supported-versions -n hypershift -o yaml
```

Hub OpenShift Container Platform version		**MCE version**				
		2.6	**2.7**	**2.8**	**2.9**	**2.10**
	Supported Hosted Control Plane OpenShift Container Platform versions	4.16 ***	4.17	4.16 to 4.18	4.17 to 4.19	4.18 to 4.20
		Is supported by MCE	**Is supported by MCE**	**Is supported by MCE**	**Is supported by MCE**	**Is supported by MCE**
4.16		Yes	Yes	Yes	Yes	Yes
4.17		Yes	Yes	Yes	Yes	Yes
4.18		No	No	Yes	Yes	Yes
4.19		No	No	No	Yes	Yes
4.20		No	No	No	Yes	Yes

***- only virtualized Hosted Control Plane is supported, and not Bare Metal Hosted Control Plane.

Prerequisites for Hosted Control Plane clusters

The prerequisites before you create virtualized or Bare Metal Hosted Control Plane clusters.

Before you begin

- To configure LVM before the installation of the IBM Fusion Fusion Data Foundation service, create the custom `configMap`. For more information about how to create, see step 11.
- To configure LVM after Fusion Data Foundation storage set up: Add drives with different capacities (compared to those used for Fusion Data Foundation storage) to any three storage nodes in the same disk slots. Ensure that the new drives are not of the same capacity as those used for the existing Fusion Data Foundation storage. Then, create a custom ConfigMap similar to the example provided in the step 11.
- Install and configure Fusion Data Foundation provider mode storage type on the hub cluster.
- If you plan to install Backup & Restore service, ensure that 12 GiB memory is available post the installation.
- The Multi-cluster IBM Fusion using Hosted Control Plane on storage is available only on platforms that support the installation of in provider mode. For more information about the support details, see Supported platforms.
- Ensure that the following ports are open on the hub cluster to deploy Hosted Control Plane clusters and support management and data traffic:

Table 1. Ports required for Fusion Data Foundation server and client

Type	Protocol	Ports	Description
Custom TCP	TCP	6789	Ceph Monitor
		3300	Ceph Monitor
		6800-7300	Ceph OSD, MGR, MDS
		9283	Ceph MGR Prometheus Exporter
		31659	API Server

Procedure

1. From the Red Hat® Operator Catalog, install MetalLB 4.16 or higher.
   Add the MetalLB Operator to your cluster so that when a service of type `LoadBalancer` is added to the cluster, MetalLB can add a fault-tolerant external IP address for the service. For the procedure to install and validate MetalLB, see https://docs.openshift.com/. Go to your specific version of OpenShift® Container Platform and check MetalLB details.
   Note: When you set up the load balancer, other applications can also get the advertised addresses. You must have enough addresses for any workloads on this cluster and created OpenShift Container Platform clusters.
2. Create MetalLB CR based on the following example:

```
apiVersion: metallb.io/v1beta1
kind: MetalLB
metadata:
  name: metallb
  namespace: metallb-system
```

3. Create an IPaddress pool.
   Note:
   - Reserve a set of unused IPs on the same CIDR as the Bare Metal network for MetalLB. The MetalLB serves these IPs to any load-balancer service that is installed on the cluster and not just the Hosted Control Plane.
   - When planning IP addresses, consider the number of Hosted Control Plane to be created and allocate additional IPs for load balancer services. Reserve three IP addresses for Fusion Data Foundation storage load balancer services. If other workloads use load balancers, factor those requirements into your IP address planning to ensure sufficient addresses are available.

   Example:

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: metallb
  namespace: metallb-system
spec:
  addresses:
  - 9.9.0.51-9.9.0.70
```

   Note: When you set up the load balancer, other applications can also get the advertised addresses. You must have enough addresses for any workloads on this cluster and created OpenShift Container Platform clusters.
4. Create `L2advertisement` based on the following example:

```
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: l2advertisement
  namespace: metallb-system
spec:
  ipAddressPools:
   - metallb
```

5. Run the following command to patch ingress:
   Note: This requirement is only for virtualized Hosted Control Plane clusters.

```
oc patch ingresscontroller -n openshift-ingress-operator default --type=json -p '[{ "op": "add", "path":
"/spec/routeAdmission", "value": {wildcardPolicy: "WildcardsAllowed"}}]'
```

6. Install Multi Cluster Engine for Kubernetes operator.
7. Create a Multi Cluster Engine CR instance based on the following example:

```
apiVersion: multicluster.openshift.io/v1
kind: MultiClusterEngine
metadata:
  name: multiclusterengine
spec:
  targetNamespace: multicluster-engine
```

   Change the parameter values according to your environment.

8. Wait until the Multi Cluster Hub instance is created, available, and in Running state.
   The Multi Cluster Engine instance must be available.
   Run the following command to check whether the Multi Cluster Engine instance is available.

   ```
   $ oc get multiclusterengine
   ```

   Example output:

   ```
   $ oc get multiclusterengine
   NAME STATUS AGE
   multiclusterengine Available 170m
   ```

9. Download hosted control plane CLI from OpenShift Container Platform console:
   a. Go to the Command Line Tools page.
   b. From the Hosted Control Plane - Hosted Control Plane Command Line Interface (CLI) section, download the CLI tar and extract from archive based on your platform.
   Note: In MCE, you can use the user interface instead of CLI. However, the CLI gives more options than the user interface.
10. To optimize `etcd` performance for hosted clusters on the hub cluster, configure LVM to use local drives on worker nodes as follows:
    a. Provision three dedicated drives on three separate worker nodes.
    b. Configure LVM on the worker nodes to manage the dedicated drives for `etcd` pods.
11. Follow the steps to configure the LVM:
    a. Create a configmap similar to the following example. This configmap is used by IBM Fusion to create a LVM cluster.

       Example configmap:

       ```
       kind: ConfigMap
       apiVersion: v1
       metadata:
         name: lvm-config
         namespace: ibm-spectrum-fusion-ns
       data:
         drives: |
           - '/dev/disk/by-path/pci-0000:61:00.0-nvme-1'
         computeNodes: |
           - compute-1-ru5.rackm01.rtp.mydomain.com
           - compute-1-ru6.rackm01.rtp.mydomain.com
           - compute-1-ru7.rackm01.rtp.mydomain.com
         nodeType: compute
       ```

       The LVM cluster uses the NVMe drive `/dev/disk/by-path/pci-0000:61:00.0-nvme-1` on the three compute nodes to provide PVCs for the `etcd` pods of the hosted clusters.

    b. To configure LVM clusters, specify the drives and nodes to be used. The `drives` field specifies the drives to be used for the LVM cluster. Use one or two drives per node. To specify two drives in a node, use the following format:

       ```
       drives: |
         - '/dev/disk/by-path/pci-0000:61:00.0-nvme-1'
         - '/dev/disk/by-path/pci-0000:63:00.0-nvme-1'
       ```

       The `computeNodes` field lists the worker nodes that provide disks for the LVM cluster. Ensure that the selected nodes are not GPU nodes.

       The `nodeType` field specifies the type of nodes to use for LVM clusters. Supported types are:
       i. **Compute**: uses disks from worker nodes.
       ii. **Control**: uses disks from all master nodes to configure the LVM cluster.
12. Increase the maximum number of pods per node from the default value of 250 to 500. For the procedure to update the value, see Managing the maximum number of pods per node .
13. For Hosted Control Plane deployments, ensure the following on the hub cluster:
    - A valid pull secret must exist in the namespace where the Hosted Control Plane cluster is defined (the namespace containing the `HostedCluster` resource).
    - The `HostedCluster` CR must include a reference to this pull secret through the `pullSecret` field.
      For example:
      Note: Credentials in pull-secret must include username, password and auth.

      ```
      pullSecret:
          name: <hosted_cluster_name>-pull-secret
      ```

14. Create a load balancer service in the `openshift-storage` namespace as follows:
    a. Define a load balancer service YAML.
       For example:

       ```
       kind: Service
       apiVersion: v1
       metadata:
         name: ocs-provider-server-load-balancer
         namespace: openshift-storage
         annotations:
           metallb.universe.tf/ip-allocated-from-pool: <address-pool-name-from-metallb>
       spec:
         ports:
           - name: provider
             protocol: TCP
             port: 50051
             targetPort: ocs-provider
             nodePort: 30756
         type: LoadBalancer
         selector:
           app: ocsProviderApiServer
       ```

    b. Apply the YAML.
       Command example:

```
oc apply -f ocs-provider-server-load-balancer.yaml
```

What to do next

1. Install an IBM Fusion hosted cluster and install Fusion Data Foundation on it. For the procedure to install the hosted cluster, see Installing the IBM Fusion base. For procedure to install Fusion Data Foundation on the hosted cluster, see Installing Fusion Data Foundation on a Hosted Control Plane cluster.

Configuring a Hosted Control Plane cluster in a proxy environment

Use these instructions to gather the required configuration and define the necessary Kubernetes resources to configure a **HostedCluster** in a proxy environment.

Before you begin

- Hosted Control Plane is deployed in a restricted environment with limited internet access.
- Proxy server is configured to provide access to external resources.

About this task

In environments where direct internet access is restricted, clusters often rely on a proxy to access required resources. This topic provides instructions on how to configure a Hosted Control Plane cluster in a proxy environment.

Procedure

1. Ensure to add the proxy configuration to the **spec** section of the **HostedCluster** CR while creating **HostedCluster** CR in hub cluster.

   ```
   spec:
     configuration:
       proxy:            # Proxy Configuration
         httpProxy: <http-proxy>
         httpsProxy: <https-proxy>
         noProxy: <list of domains to not route through proxy>
         trustedCA:
           name: <trusted-ca-secret>
   ```

   The following proxy configuration settings are required:
   - **httpProxy**: The HTTP proxy URL.
   - **httpsProxy**: The HTTPS proxy URL.
   - **noProxy**: A list of domains that should not be routed through the proxy.
   - **trustedCA**: The name of the trusted CA secret.
     Note: The filed **trustedCA** should be empty if no custom certificate is used in proxy server.
2. To retrieve the existing proxy configuration values, run the following command on the Hub cluster:

   ```
   oc get proxy cluster -o yaml
   ```

   Example output:

   ```
   apiVersion: config.openshift.io/v1
   kind: Proxy
   metadata:
     name: cluster
     resourceVersion: '532'
   spec:
     httpProxy: '<http://squid:squid@172.x.y.z:3128'>
     httpsProxy: '<http://squid:squid@172.x.y.z:3128'>
     noProxy: '.mydomain.com,9999,metal3-state.openshift-machine-api,5050,6385,8089,6180,80,8084,169.253.0.0/16'
     trustedCA:
       name: ''
   status:
     httpProxy: '<http://squid:squid@172.x.y.z:3128'>
     httpsProxy: '<http://squid:squid@172.x.y.z:3128'>
     noProxy:
   '.cluster.local,.mydomain.com,.svc,10.128.0.0/14,127.0.0.1,169.253.0.0/16,172.a.b.c/24,172.x.y.z/16,5050,6180,6385,80,8084
   ,8089,9999,api-int.isfdelace.mydomain.com,localhost,metal3-state.openshift-machine-api'
   ```

3. Ensure that the following additional settings are added to the **noProxy** field in the **HostedCluster** CR while creating the CR:
   - **localhost**: **127.0.0.1**
   - Cluster subdomain (For example, **isfdelace.mydomain.com**)
   - Cluster base domain (For example, **.mydomain.com**)
   - Service URL extension: **.svc**
   - Service network and cluster network of the base Cluster
   - Service network and cluster network of the HCP cluster
   - First IP of the service network from the HCP
4. To retrieve the cluster network and service network values, run the following command on the hub cluster:

   ```
   oc get network cluster -o yaml
   ```

5. When adding hosts to a Bare Metal hosted control plane cluster, you must download the **discovery.iso** from the MCE user interface. When selecting Add Hosts > With Discovery ISO, click Show proxy settings and add the following values:
   - HTTP proxy URL
   - HTTPS proxy URL
   - No proxy domains

These field values must match the values used in the earlier steps.

Deployment of virtualized clusters with Fusion Data Foundation

Create virtualized (Hosted Control Plane) clusters with the Fusion Data Foundation as storage type.

- **Planning and prerequisites for your virtualized clusters**
  The prerequisites before you create Virtualization clusters.
- **Creating a virtualized Hosted Control Plane cluster**
  Create an OpenShift virtualization Hosted Control Plane cluster.

Planning and prerequisites for your virtualized clusters

The prerequisites before you create Virtualization clusters.

- From the Red Hat® Operator Catalog, deploy Red Hat OpenShift® Virtualization. Use the latest version available in your Red Hat operator catalog as the version is driven by underlying the OpenShift Container Platform version. For more information about the supported versions, see Interoperability matrix.
  It is a software component of OpenShift Container Platform to run and manage virtual machine workloads alongside container workloads. To validate the installation, see Red Hat Documentation.
- Create hyperconverged CR based on the following example:

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec: {}
```

- Wait for five to ten minutes for the CR status to be available and the OpenShift Container Platform console to refresh and show up a new Virtualization menu.

Creating a virtualized Hosted Control Plane cluster

Create an OpenShift® virtualization Hosted Control Plane cluster.

Before you begin

- The IBM Fusion with the Multi-Cluster Engine version 2.7 or higher supports hosted clusters, which run on OpenShift version 4.16. For more information about the supported versions, see Interoperability matrix. The upgrade of OpenShift Container Platform is supported on Hosted Control Plane clusters. For creating a hosted cluster with KubeVirt platform by using CLI, see Red Hat Documentation.
- Ensure you meet the following prerequisites:
  - Prerequisites for Hosted Control Plane clusters
  - Planning and prerequisites for your virtualized clusters

About this task

If the virtual cluster is not created or has issues, see Troubleshooting hosted control planes.

Procedure

1. Log in to OpenShift Container Platform console, and go to Multi Cluster Engine (MCE) view.
2. Go to Infrastructure > Clusters.
3. In the Cluster list tab, click Create cluster.
4. Select Red Hat OpenShift Virtualization as your infrastructure provider.
5. In the Control plane type, select Hosted. A Create cluster wizard page gets displayed.
6. In the Create cluster wizard, enter your cluster details:
   a. In the Cluster details wizard page, enter your cluster details.

      Release image
      : Select your OpenShift version. For more information about the supported versions, see Interoperability matrix.
      Note: No multi arch images must be used while provisioning Hosted Control Plane clusters.

      Additional labels
      : Enter `isf.ibm.com/fusion-base=`, `isf.ibm.com/fusion-fdf=`, and `isf.ibm.com/fusion-backup=` labels to install IBM Fusion, Fusion Data Foundation, and Backup & Restore service on the newly created cluster.

      Pull secret
      : A pull secret is required for the creation of a Hosted Control Plane cluster. Include the `cp.icr.io` for the Fusion services.

   b. Click Next to go to the Node pools page of the wizard.
   c. In the Node pools page, enter the location and size of the nodes.
   d. Click Next to go to the Review and create page of the wizard.
   e. In the Review and create page, go through the details and click Create.

Alternatively, use the following command line example to create a Hosted Control Plane for OpenShift Virtualization by using the KubeVirt platform type.
Note: Ensure you check the following points before you proceed:

- Ensure you export the variables used in the example with their right values.
- Also, check whether you have the latest Hosted Control Plane Command Line interface (CLI). If you do not have the latest version, go to Red Hat® OpenShift Container Platform console and download the latest version from Command line tools page.

```
# Set environment variables
export CLUSTER_NAME=example
export PULL_SECRET="$HOME/pull-secret"
export MEM="6Gi"
export CPU="2"
export WORKER_COUNT="2"

hcp create cluster kubevirt \
 --name $CLUSTER_NAME \
 --node-pool-replicas $WORKER_COUNT \
 --pull-secret $PULL_SECRET \
 --memory $MEM \
 --cores $CPU \
 --etcd-storage-class lvms-hcp-etcd \
 --infra-availability-policy HighlyAvailable \
 --root-volume-storage-class ocs-storagecluster-ceph-rbd-virtualization \
 --root-volume-volume-mode Block \
--root-volume-access-modes ReadWriteMany
```

Run the following command to get a list of available parameters:

```
hcp create cluster kubevirt --help
```

7. To validate the clusters, do the following steps:
   a. Log in to the OpenShift Container Platform console of the hub, and go to multi cluster view.
   b. Go to Infrastructure > Clusters.
   c. From the Clusters list, click a hosted cluster name link to view the details.

Note: Hosted Control Plane cluster creation can take more than 30 minutes.

What to do next

1. Install the IBM Fusion base on the Hosted Control Plane cluster. For instructions, see Installing the IBM Fusion base.
2. Install the Fusion Data Foundation storage client on the Hosted Control Plane cluster. For instructions, see Installing Fusion Data Foundation on a Hosted Control Plane cluster.

Deploying Bare Metal Hosted Control Plane cluster in disconnected environment

A disconnected environment deployment operates without internet access. You can deploy Hosted Control Plane (HCP) in a disconnected environment on Bare Metal infrastructure.

For the procedure to deploy Bare Metal Hosted Control Plane cluster in disconnected network, see Red Hat documentation.

Important: While you follow the Red Hat documentation, take note of the following two steps that must be done differently:

1. The pull secret for Hosted Control Plane in its namespace must reference the host only, without including the repository path.
2. Skip *Deploying a registry for hosted control planes in a disconnected environment* procedure. See Red Hat documentation.
3. When you create the hosted cluster CR, remove OIDC service.

```
- service: OIDC
  servicePublishingStrategy:
    type: Route
```

4. Skip *Creating bare metal hosts for the hosted cluster* procedure. See Red Hat documentation.
5. When creating an infrastructure environment to configure host inventory, select the following network types: Static, Bond, and Bridge.

Examples of CRs and CMs created:

Configuring `AgentServiceConfig` resources

1. Create a configmap named `custom-registries` in the `multicluster-engine` namespace.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: custom-registries
  namespace: multicluster-engine
data:
  <registry_name>..<port>: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  <registry_name>..<port>: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  <registry_name>..<port>: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  registries.conf: |
    unqualified-search-registries = ["registry.access.redhat.com", "docker.io"]
    short-name-mode = ""
```

```
    [[registry]]
       prefix = ""
       location = "quay.io/openshift-release-dev/ocp-v4.0-art-dev"
       mirror-by-digest-only = true
       [[registry.mirror]]
         location = "$TARGET_PATH/openshift/release"

    [[registry]]
       prefix = ""
       location = "quay.io/openshift-release-dev/ocp-release"
       mirror-by-digest-only = true
       [[registry.mirror]]
         location = "$TARGET_PATH/openshift/release-images"

    [[registry]]
       prefix = ""
       location = "registry.redhat.io/multicluster-engine"
       mirror-by-digest-only = true
       [[registry.mirror]]
         location = "$TARGET_PATH/multicluster-engine"

    [[registry]]
       prefix = ""
       location = "registry.redhat.io/openshift4"
       mirror-by-digest-only = true
       [[registry.mirror]]
         location = "$TARGET_PATH/openshift4"

    [[registry]]
       prefix = ""
       location = "registry.redhat.io/rhel9"
       mirror-by-digest-only = true
       [[registry.mirror]]
         location = "$TARGET_PATH/rhel9"

    [[registry]]
       prefix = ""
       location = "registry.redhat.io/rhel8"
       mirror-by-digest-only = true
       [[registry.mirror]]
         location = "$TARGET_PATH/rhel8"

    [[registry]]
       prefix = ""
       location = "registry.redhat.io/redhat"
       mirror-by-digest-only = true
       [[registry.mirror]]
         location = "$TARGET_PATH/redhat"
```

Note:

- Replace `<registry_name>..<port>` with the actual registry name and port.
- Provide the CA root certificate and intermediate certificate of the registry.
- The `unqualified-search-registries` is mandatory.
- Mirrors list can be found in the workspace folder created when mirroring OpenShift® Container Platform and multi-cluster engine or `metalLB` images.

2. Create an `AgentServiceConfig` resource named `agent`.

```
apiVersion: agent-install.openshift.io/v1beta1
kind: AgentServiceConfig
metadata:
  name: agent
spec:
  databaseStorage:
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: 200Gi
    storageClassName: ocs-storagecluster-ceph-rbd
  filesystemStorage:
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: 200Gi
    storageClassName: ocs-storagecluster-ceph-rbd
  imageStorage:
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: 100Gi
  mirrorRegistryRef:
    name: custom-registries
  osImages:
    - cpuArchitecture: x86_64
      openshiftVersion: 4.18.20
      rootFSUrl: 'http://172.20.x.y:8118/hcp/images/rhcos-418.94.202501221327-0-live-rootfs.x86_64.img'
      url: 'http://172.20.x.y:8118/hcp/images/rhcos-418.94.202501221327-0-live.x86_64.iso'
      version: 418.94.202501221327-0
```

Note:

- Specify the configmap name created earlier in the `multicluster-engine` namespace under `mirrorRegistryRef`.
- Add one entry for each OpenShift Container Platform version you wish to deploy hosted cluster with under `osImages`.
- The `rootFSUrl` and `url` are links to files extracted earlier and hosted on a web server.

- The `version` is the core OS version of the downloaded ISO.

Configuring TLS certificates for a disconnected installation of Hosted Control Plane

Create a configmap named `registry-config` in the `openshift-config` namespace.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: registry-config
  namespace: openshift-config
data:
   <registry_name>..<port>: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  <registry_name>..<port>: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  <registry_name>..<port>: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
```

Note:

- Replace `<registry_name>..<port>` with the actual registry name and port.
- Provide the CA root certificate and intermediate certificate of the registry.

Creating a `hostedcluster` on Bare Metal

1. Create two namespaces: `fusion-bm` and `fusion-bm-fusion-bm`.

   ```
   apiVersion: v1
   kind: Namespace
   metadata:
     name: fusion-bm-fusion-bm
   spec: {}
   ```

   ```
   apiVersion: v1
   kind: Namespace
   metadata:
     name: fusion-bm
   spec: {}
   ```

2. Create a configmap named `user-ca-bundle` in the `fusion-bm` namespace.

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: user-ca-bundle
     namespace: fusion-bm
   data:
     ca-bundle.crt: |
       -----BEGIN CERTIFICATE-----
       -----END CERTIFICATE-----
   ```

   Note: Provide the registry CA root certificate.

3. Create a secret named `fusion-bm-pull-secret` and `sshkey-cluster-fusion-bm` in the `fusion-bm` namespace.

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: fusion-bm-pull-secret
     namespace: fusion-bm
   data:
     .dockerconfigjson: xxxxxxxxx
   type: kubernetes.io/dockerconfigjson
   ---
   apiVersion: v1
   kind: Secret
   metadata:
     name: sshkey-cluster-fusion-bm
     namespace: fusion-bm
   data:
     id_rsa.pub: xxxxxxxxx
   type: Opaque
   ```

4. Create a role named `capi-provider-role` in the `fusion-bm-fusion-bm` namespace.

   ```
   apiVersion: rbac.authorization.k8s.io/v1
   kind: Role
   metadata:
     name: capi-provider-role
     namespace: fusion-bm-fusion-bm
   rules:
     - verbs:
         - '*'
       apiGroups:
         - agent-install.openshift.io
       resources:
         - agents
   ```

5. Create a `hostedcluster` CR named `fusion-bm` in the `fusion-bm` namespace.

```
apiVersion: hypershift.openshift.io/v1beta1
kind: HostedCluster
metadata:
  name: fusion-bm
  namespace: fusion-bm
spec:
  fips: false
  release:
    image: $TARGET_PATH/openshift/release-images:4.18.20-x86_64
  dns:
    baseDomain: xyz.com
  controllerAvailabilityPolicy: HighlyAvailable
  etcd:
    managed:
      storage:
        persistentVolume:
          size: 8Gi
          storageClassName: ocs-storagecluster-ceph-rbd
        type: PersistentVolume
    managementType: Managed
  infrastructureAvailabilityPolicy: SingleReplica
  platform:
    agent:
      agentNamespace: fusion-bm-fusion-bm
    type: Agent
  additionalTrustBundle:
    name: user-ca-bundle
  networking:
    clusterNetwork:
      - cidr: 10.132.0.0/14
    networkType: OVNKubernetes
    serviceNetwork:
      - cidr: 172.31.0.0/16
  pullSecret:
    name: fusion-bm-pull-secret
  configuration:
    operatorhub:
      disableAllDefaultSources: true
  capabilities: {}
  sshKey:
    name: sshkey-cluster-fusion-bm
  autoscaling: {}
  imageContentSources:
    - mirrors:
        - $TARGET_PATH/openshift/release
      source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
    - mirrors:
        - $TARGET_PATH/redhat
      source: registry.redhat.io/redhat
    ...
  olmCatalogPlacement: management
  services:
    - service: APIServer
      servicePublishingStrategy:
        type: LoadBalancer
    - service: OAuthServer
      servicePublishingStrategy:
        type: Route
    ...
```

Note:

- Specify the configmap name created earlier in the **fusion-bm** namespace under **additionalTrustBundle**.
- Specify the secret name created earlier in the **fusion-bm** namespace under **pullSecret** and **sshKey**.
- Specify the image content sources and mirrors.

6. Wait for the **hostedcluster** CR to get created and the API to become available.
7. Create a **NodePool** CR named **fusion-bm-np** in the **fusion-bm** namespace.

```
apiVersion: hypershift.openshift.io/v1beta1
kind: NodePool
metadata:
  name: fusion-bm-np
  namespace: fusion-bm
spec:
  arch: amd64
  clusterName: fusion-bm
  management:
    autoRepair: false
    replace:
      rollingUpdate:
        maxSurge: 1
        maxUnavailable: 0
      strategy: RollingUpdate
    upgradeType: InPlace
  nodeDrainTimeout: 0s
  platform:
    agent:
      agentLabelSelector:
        matchLabels:
          inventory.agent-install.openshift.io/hcp: fusion-bm
    type: Agent
  release:
    image: '$TARGET_PATH/openshift/release-images:4.18.20-x86_64'
  replicas: 3
```

Note:

- Specify the cluster name and release image.
- Specify the number of replicas.

Create a host inventory

1. Create a secret named `pullsecret-fusion-bm` in the `fusion-bm-fusion-bm` namespace.

```
apiVersion: v1
kind: Secret
metadata:
  name: pullsecret-fusion-bm
  namespace: fusion-bm-fusion-bm
data:
  .dockerconfigjson: xxxxxxxx
type: kubernetes.io/dockerconfigjson
```

2. Create an `InfraEnv` CR named `fusion-bm` in the `fusion-bm-fusion-bm` namespace.

```
apiVersion: agent-install.openshift.io/v1beta1
kind: InfraEnv
metadata:
  name: fusion-bm
  namespace: fusion-bm-fusion-bm
  labels:
    agentclusterinstalls.extensions.hive.openshift.io/location: us
    networkType: static
spec:
  agentLabels:
    agentclusterinstalls.extensions.hive.openshift.io/location: us
  imageType: minimal-iso
  mirrorRegistryRef:
    name: custom-registries
    namespace: multicluster-engine
  ipxeScriptType: DiscoveryImageAlways
  cpuArchitecture: x86_64
  sshAuthorizedKey: xxxxxxxx
  pullSecretRef:
    name: pullsecret-fusion-bm
  nmStateConfigLabelSelector:
    matchLabels:
      infraenvs.agent-install.openshift.io: fusion-bm
```

Note:

- Specify the mirror registry reference and namespace.
- Specify the image type and CPU architecture.
- Specify the SSH authorized key and pull secret reference.

Deploying virtualized Hosted Control Plane cluster in disconnected environment

A disconnected environment deployment operates without internet access. You can deploy Hosted Control Plane (HCP) in a disconnected environment on the virtualized infrastructure.

For the procedure to deploy virtualized Hosted Control Plane cluster in disconnected network, see Red Hat documentation.

Important: While you follow the Red Hat documentation, take note of the following two steps that must be done differently:

1. The pull secret for the Hosted Control Plane in its namespace must only reference the host. Do not include the repository path.
2. When you create the virtualized Hosted Control Plane cluster CR, remove OIDC service.

```
- service: OIDC
  servicePublishingStrategy:
    type: Route
```

Examples of CRs and CMs created while deploying virtualized Hosted Control Plane:

1. Create two namespaces: `clusters` and `clusters-hosted-cluster-name`.
   For example, `clusters-disconnected1example`

```
apiVersion: v1
kind: Namespace
metadata:
  creationTimestamp: null
  name: clusters-disconnected1
spec: {}
status: {}
```

```
apiVersion: v1
kind: Namespace
metadata:
  creationTimestamp: null
  name: clusters
spec: {}
status: {}
```

2. Copy the registry CA cert and create a configmap in the `clusters` namespace.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: user-ca-bundle
  namespace: clusters
data:
  ca-bundle.crt: |
    // Registry1 CA
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----

    // Registry2 CA
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----

    // Registry3 CA
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
```

3. Create a pull secret in the `clusters` namespace.

```
kind: Secret
apiVersion: v1
metadata:
  name: disconnected-secret
  namespace: clusters
data:
  .dockerconfigjson: xxxxxxxx
type: kubernetes.io/dockerconfigjson
```

4. Create a `hostedcluster` CR in the `clusters` namespace.

```
apiVersion: hypershift.openshift.io/v1beta1
kind: HostedCluster
metadata:
  name: disconnected1
  namespace: clusters
spec:
  fips: false
  release:
    image: $TARGET_PATH/openshift/release-images:4.18.20-x86_64
  dns:
    baseDomain: xyz.com
  controllerAvailabilityPolicy: HighlyAvailable
  etcd:
    managed:
      storage:
        persistentVolume:
          size: 8Gi
          storageClassName: ocs-storagecluster-ceph-rbd
        type: PersistentVolume
    managementType: Managed
  infrastructureAvailabilityPolicy: SingleReplica
  platform:
    kubevirt:
      baseDomainPassthrough: true
    type: KubeVirt
  additionalTrustBundle:
    name: user-ca-bundle
  networking:
    clusterNetwork:
      - cidr: 10.132.0.0/14
    networkType: OVNKubernetes
    serviceNetwork:
      - cidr: 172.31.0.0/16
  pullSecret:
    name: disconnected-secret
  configuration:
    operatorhub:
      disableAllDefaultSources: true
  capabilities: {}
  sshKey:
    name: ''
  autoscaling: {}
  imageContentSources:
    - mirrors:
        - $TARGET_PATH/openshift/release
      source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
    - mirrors:
        - $TARGET_PATH/redhat
      source: registry.redhat.io/redhat
    - mirrors:
        - $TARGET_PATH/rhel9
      source: registry.redhat.io/rhel9
    - mirrors:
        - $TARGET_PATH/rhel8
      source: registry.redhat.io/rhel8
    - mirrors:
        - $TARGET_PATH/openshift/release-images
      source: quay.io/openshift-release-dev/ocp-release
    - mirrors:
        - $TARGET_PATH/openshift4
      source: registry.redhat.io/openshift4
    - mirrors:
        - $TARGET_PATH/multicluster-engine
      source: registry.redhat.io/multicluster-engine
```

```
  olmCatalogPlacement: management
  services:
    - service: APIServer
      servicePublishingStrategy:
        type: LoadBalancer
    - service: Ignition
      servicePublishingStrategy:
        type: Route
    - service: Konnectivity
      servicePublishingStrategy:
        type: Route
    - service: OAuthServer
      servicePublishingStrategy:
        type: Route
```

Note: While you follow the Red Hat documentation, take note of the following two steps that must be done differently:

a. The pull secret for the Hosted Control Plane in its namespace must only reference the host. Do not include the repository path.

b. When you create the virtualized Hosted Control Plane cluster CR, remove OIDC service.

```
- service: OIDC
  servicePublishingStrategy:
    type: Route
```

5. Create a `NodePool` CR in the `clusters` namespace.

```
apiVersion: hypershift.openshift.io/v1beta1
kind: NodePool
metadata:
  name: disconnected1
  namespace: clusters
spec:
  arch: amd64
  clusterName: disconnected1
  management:
    autoRepair: false
    upgradeType: Replace
  nodeDrainTimeout: 0s
  nodeVolumeDetachTimeout: 0s
  platform:
    kubevirt:
      attachDefaultNetwork: true
      compute:
        cores: 2
        memory: 6Gi
      networkInterfaceMultiqueue: Enable
      rootVolume:
        persistent:
          size: 32Gi
        type: Persistent
    type: KubeVirt
  release:
    image: $TARGET_PATH/openshift/release-images:4.18.20-x86_64
  replicas: 2
```

Note:

- Specify the cluster name and release image.
- Specify the number of replicas along with configuration such as cores, memory and so on.

Deployment of Bare Metal clusters with Fusion Data Foundation

Create Bare Metal hosted clusters with Fusion Data Foundation as the storage type. It supports both Bare Metal internal servers in a rack and Bare Metal external servers.

The following are the high-level steps to create a Bare Metal Hosted Control Plane cluster:
Note: Ensure you meet all the prerequisites before you deploy:

- Prerequisites for Hosted Control Plane clusters
- Planning and prerequisites for your Bare Metal Hosted Control Plane

1. Plan for the Bare Metal Hosted Control Plane
2. Create the Infrastructure Environment.
3. Import the hosts by using Multi Cluster Engine (MCE):
   a. Create the NMStateConfig for each host.
   b. Download the `discovery.iso`.
   c. Boot the host with the `discovery.iso`.
   d. Accept host into MCE inventory.
4. Create the Hosted Control Plane cluster.
5. Configuring ingress for newly created Bare Metal hosted cluster.

- **Planning and prerequisites for your Bare Metal Hosted Control Plane**
  As a prerequisite, gather networking information of the hosting cluster. Define and apply DHCP and DNS information, and configure host inventory settings.
- **Deploying Bare Metal clusters with Fusion Data Foundation**
  You can create a Hosted Control Plane cluster by using Bare Metal hosts either through the OpenShift Console or the Hosted Control Plane CLI.
- **Configuring ingress for Bare Metal Hosted cluster**
  After you deploy the Bare Metal clusters with Fusion Data Foundation, you must configure the ingress for you to view the changes in the user interface.
- **Expanding your Hosted Control Plane Bare Metal cluster**
  Add a new host to the exiting Bare Metal Hosted Control Plane cluster.

Planning and prerequisites for your Bare Metal Hosted Control Plane

As a prerequisite, gather networking information of the hosting cluster. Define and apply DHCP and DNS information, and configure host inventory settings.

Plan for the Bare Metal Hosted Control Plane:

1. Choose your configuration.
2. Gather networking information about IBM Fusion HCI hosting cluster.
3. Define the new networking information.
4. Apply the networking information to the DNS and DHCP in your environment

Choose your configuration

There are two configurations of Bare Metal Hosted Control Plane cluster.

Internal
: In the internal Bare Metal Hosted Control Plane, the servers that are used in the cluster are within the IBM Fusion HCI. When the servers are internal to the IBM Fusion HCI, most of the network information required is collected and present in the Custom Resources of the IBM Fusion HCI.

External
: In external Bare Metal Hosted Control Plane, the servers used in the cluster are external to the IBM Fusion HCI. For external servers, you must gather the network information from your lab network administrator.
Connecting your external servers to IBM Fusion HCI:

  External servers must be connected to the IBM Fusion HCI through a switch to allow connectivity. These servers can then be imported into Multi Cluster Engine (MCE) as part of a Host Inventory. For instructions to import and its usage in a Hosted Control Plane, see Deploying Bare Metal clusters with Fusion Data Foundation.

Networking prerequisites

Step 1: Gather network information
: The networking information consists of the entries that the network administrator needs to add to the network. You need the following information about the host cluster:

  Keep the following data ready to use when you create a cluster and `nmstate` config for each node.

  Cluster wide information:
  Note: The cluster name, sub domain, and domain are all for one hosted cluster. Create them for each of your hosted clusters.

Details	Description	Example values
Base Domain	Cluster base domain you want to use for Hosted Control Plane cluster to be created	`mydomain.com`
Cluster name	Cluster name you want to use for HCP cluster to be created.	`fusion-bm`
Sub Domain	This is always <cluster name>.<base domain>	`fusion-bm.mydomain.com`

  Individual nodes in the cluster:

Entry	Description	Example values
BMC address	This is value of field named ipv6ULA or ipv4 in the kickstart CM. You can use ipv6ULA or ipv4.	Example for ipv6ULA - fd9c:316d:179e:c0de:a94:efff:fefe:2ec1 Example value for ipv4 - 170.254.2.9
IMM USER/Password	To get this value, search for `secretName` field in the `kickstart config`. Inside that secret, search for `defaultUserName` and `defaultUserPasswrd` fields for IMM user and Password respectively.	
Bare metal primary interface IP	Unused IP that you plan yo use for Bare Metal host in Hosted Control Plane cluster	172.17.x.y
Macaddress	It is available in the `kickstart config networkInterfaces`, the bond0 macAddress. You need both Bare Metal Primary and secondary interface Mac addresses. You can find the macaddress in the slot section that includes the first macaddress. This guidance is for internal HCI managed nodes. Consult your network administrator for external bare metal nodes	
Gateway IP	Run the following command to get this from one of HCI nodes: `ip r |grep default|grep br-ex|awk '{print $3}'` If the Hosted Control Plane nodes use the same gateway server as the base HCI cluster. Otherwise, get it from the network administrator. It is only needed when you use the static IP assignment for nodes in Hosted Control Plane cluster. For DHCP managed nodes, this field is not needed.	172.17.x.1
MTU	This can be 9000 or 1500 based on base HCI rack Bare metal MTU used during installation. Find this value from field `mtuCount` in secret `userconfig-secret` in namespace `ibm-spectrum-fusion-ns`.	9000
Bare metal Primary interface's first port's Mac address	This is the value of primary macaddress in kickstart config map for a given node where the interfaceType is baremetal.	08:c0:eb:d4:16:4e
Bare metal Primary interface's second port's Mac address	This is the value of secondary macaddress in kickstart config map for given node where interfaceType is baremetal.	08:c0:eb:d4:16:4f

Entry	Description	Example values
Bare metal Primary interface's Mac address	This is value of field **networkInterfaces** -> macAddress in kickstart config map for a given node where the **interfaceType** is baremetal.	08:c0:eb:d4:16:4e
Bare metal Primary interface's (bond0) first port/interface name	This is value of field networkInterfaces -> interfaceLeg1 in kickstart config map for a given node, where interfaceType is baremetal.	ens3f0np0
Bare metal Primary interface's (bond0) second port/interface name	This is value of field networkInterfaces -> interfaceLeg2 in kickstart config map for given node, where interfaceType is baremetal.	

Example servers external to IBM Fusion HCI:

Server	BMC address of external Bare Metal node	HCP Cluster name	BMC address	Switch bond0 info	mac Address	Type
sr650immru22 tc11-m04-ru22.mydomain.com	fd8c:215d:178e:c0de:3a68:ddff:fe57:2e95	extbarehcp01m04	1.23.45.178	ens1f0np0 b8:3f:d2:3c:6d:60 ens1f1np1 b8:3f:d2:3c:6d:61	b8:3f:d2:3c:6d:60	External

Example servers internal to IBM Fusion HCI:

Server	macaddress Example	HCP Cluster name	BMC address	BMC address	Type/ Label
compute-1-ru8	1070FDB8DF72	fusion-bm	1.23.45.163	fd8c:215d:178e:c0de:a94:efff:fefd:e7e1	Compute

DHCP update for the cluster:

Note: This section is needed only when you plan to use DHCP to manage IP and hostname assignment for nodes in Hosted Control Plane. If you use static IP assignment, then skip this section.
You need DHCP only if you want to manage your server IPs using the DHCP for hosted clusters.
To ensure that the individual servers have an IP address that can be contacted by the Hosted Control Plane cluster, make an entry for each host that links its macaddress to an IP address that can be contacted by the IBM Fusion HCI cluster. The macaddress can be found on the back of each physical server.

Example:

Server	macaddress	IP address
compute-1-ru8.fusion-bm.mydomain.com	10:70:FD:B8:DF:72	172.17.x.y

DNS update for Cluster:

Note: Add DNS entry of every host user who wants to create Bare Metal Hosted Control Plane cluster.
The DNS update table entries for each Hosted Control Plane cluster that links a lab provided Server IP address to the ***.apps.NameofCluster.FQDN**. This IP address is used to create a load balancer on the Hosted Control Plane cluster to allow ingress to that cluster.

Load balancer IP	Entry for DNS	FQDN Host	Ingress Type	HCP Clustername	Domain name
1.23.45.910	*.apps.barehcp01m04.mydomain.com	compute-1-ru8.fusion-bm.mydomain.com	loadbalancer	fusion-bm	mydomain.com

Server name	Entry for DNS	IP address
compute-1-ru8	compute-1-ru8.fusion-bm.mydomain.com	172.17.x.y

Step 2: Apply the networking information to the DNS and DHCP in your environment
After you gather hosting cluster information, apply the following in your environment (lab network environment):

- DHCP update
- DNS update for host

Multi Cluster Engine (MCE)

Step 1: Configure Host Inventory Settings
After the initial setup, configure the host inventory settings. For more information about the host inventory settings, see Red Hat documentation.

Step 2: Create project in host IBM Fusion OpenShift Container Platform cluster for host inventory for Hosted Control Plane cluster to be created
From the OpenShift Container Platform console of IBM Fusion, create a project designated for the host inventory along with other necessary resources to support the Hosted Control Plane cluster.
You can use any name. In this example, cluster name is used for simple identification.

Step 3: Create your infrastructure environment
The next step is to create an infrastructure environment in MCE or using the Hosted Control Plane CLI. This infrastructure environment contains the list of Bare Metal hosts that you can select to create a Bare Metal Hosted Control Plane.
Note: Create one infrastructure environment per hosted control plane cluster. Each Hosted Control Plane can only add hosts from a single infrastructure.

1. In the MCE user interface, go to Infrastructure > Host Inventory.
2. Select Create infrastructure environment.
3. On the Create infrastructure environment page, enter the following details:

Name
Name of the infrastructure. No specific name is required.

Network type
Select Static IP, Bridges, and Bonds.

Location
Location of hosts

Labels
Optional value.

Pull Secret

The pull secret must include the following:

- cloud.openshift.com
- cp.icr.io
- quay.io
- registry.connect.redhat.com
- registry.redhat.io

SSH public key

Generate a private public ssh key pair from bastion host and specify public key in ssh field. It is optional but recommended.

Step 4: Import the hosts by using MCE

With the infrastructure environment created, the next steps are to boot individual servers with an ISO to import them into the host inventory.

Step 4.1: Create the NMStateConfig for each host

The **NMStateConfig** allows the server to be recognized by the IBM Fusion HCI cluster. For additional information, see Red Hat documentation.
The NMStateConfig varies based on whether static IP assignment or DHCP IP assignment is used. For a fresh IBM Fusion HCI 2.10 installation or an upgrade from 2.9 to 2.10 with the network migration script applied, use the new NMStateConfig. If IBM Fusion HCI is upgraded from version 2.9 to 2.10 without applying the network migration script, retain the old NMStateConfig configuration.

Static IP

New **NMStateConfig** for static IP assignment:

```
apiVersion: agent-install.openshift.io/v1beta1
kind: NMStateConfig
metadata:
  labels:
    infraenvs.agent-install.openshift.io: <same as infra name>
  name: <any unique name for host within infra for example fusion-bm-ru8>
  namespace: <same namespace as used for infra>
spec:
  config:
    dns-resolver:
      config:
        server:
          - <DNS server IP>
    interfaces:
      - ipv6:
          enabled: false
        link-aggregation:
          mode: 802.3ad
          options:
            lacp_rate: '1'
            miimon: '140'
            xmit_hash_policy: '1'
          ports:
            - <Bare metal Primary interface's first port/interface name>
            - <Bare metal Primary interface's second port/interface name>
        mac-address: <Bare metal Primary interface's first port's Mac address>
        mtu: <MTU value>
        name: bond0
        state: up
        type: bond
      - ipv4:
          address:
            - ip: <Bare metal primary interface IP>
              prefix-length: 24
          dhcp: false
          enabled: true
        ipv6:
          enabled: false
        name: <bond name followed by VLAN ID of type OpenShift Customer VLAN present on the cluster for example
bond0.VLAN_ID>
        state: up
        type: vlan
        vlan:
          base-iface: bond0
          id: <VLAN ID of type OpenShift Customer VLAN present on the cluster>
    routes:
      config:
        - destination: 0.0.0.0/0
          next-hop-address: <Gateway IP>
          next-hop-interface: <bond name followed by VLAN ID of type OpenShift Customer VLAN present on the cluster
for example bond0.VLAN_ID>
  interfaces:
    - macAddress: <Bare metal Primary interface's first port's Mac address>
      name: Bare metal Primary interface's first port/interface name>
    - macAddress: <Bare metal Primary interface's second port's Mac address>
      name: Bare metal Primary interface's second port/interface name>
```

Old **NMStateConfig** for static IP Assignment:

```
apiVersion: agent-install.openshift.io/v1beta1
kind: NMStateConfig
metadata:
  labels:
    infraenvs.agent-install.openshift.io: <same as infra name>
  name: <any unique name for host within infra for example fusion-bm-ru8>
  namespace: <same namespace as used for infra>
spec:
  config:
    dns-resolver:
      config:
        server:
```

```
          - <DNS server IP>
    interfaces:
      - ipv4:
          address:
            - ip: <Bare metal primary interface IP>
              prefix-length: 24
          dhcp: false
          enabled: true
        ipv6:
          enabled: false
        link-aggregation:
          mode: 802.3ad
          options:
            lacp_rate: '1'
            miimon: '140'
            xmit_hash_policy: '1'
          ports:
            - <Bare metal Primary interface’s first port/interface name>
            - <Bare metal Primary interface’s second port/interface name>
        mac-address: <Bare metal Primary interface’s first port’s Mac address>
        mtu: <MTU value>
        name: bond0
        state: up
        type: bond
    routes:
      config:
        - destination: 0.0.0.0/0
          next-hop-address: <Gateway IP>
          next-hop-interface: bond0
  interfaces:
    - macAddress: <Bare metal Primary interface’s first port’s Mac address>
      name: <Bare metal Primary interface’s first port/interface name>
    - macAddress: <Bare metal Primary interface’s second port’s Mac address>
      name: <Bare metal Primary interface’s second port/interface name>
```

DHCP

New `NMStateConfig` for DHCP IP assignment:

```
apiVersion: agent-install.openshift.io/v1beta1
kind: NMStateConfig
metadata:
  labels:
    infraenvs.agent-install.openshift.io: <same as infra name>
  name: <any unique name for host within infra for example fusion-bm-ru8>
  namespace: <same namespace as used for infra>
spec:
  config:
    interfaces:
     - ipv6:
        enabled: false
       link-aggregation:
        mode: 802.3ad
        options:
          lacp_rate: "1"
          miimon: "140"
          xmit_hash_policy: "1"
        ports:
        - <Bare metal Primary interface's first port/interface name>
        - <Bare metal Primary interface's second port/interface name>
      mac-address: <Bare metal Primary interface's first port's Mac address>
      mtu: <MTU value>
      name: bond0
      state: up
      type: bond
     - ipv4:
          dhcp: true
          enabled: true
        ipv6:
          enabled: false
        name: <bond name followed by VLAN ID of type OpenShift Customer VLAN present on the cluster. For example
bond0.VLAN_ID>
        state: up
        type: vlan
        vlan:
          base-iface: bond0
          id: <VLAN ID of type OpenShift Customer VLAN present on the cluster>
  interfaces:
    - macAddress: <Bare metal Primary interface's first port's Mac address>
      name: <Bare metal Primary interface's first port/interface name>
    - macAddress: <Bare metal Primary interface's second port's Mac address>
      name: <Bare metal Primary interface's second port/interface name>
```

Old NMStateConfig:

```
apiVersion: agent-install.openshift.io/v1beta1
kind: NMStateConfig
metadata:
  labels:
    infraenvs.agent-install.openshift.io: <same as infra name>
  name: <any unique name for host within infra for example fusion-bm-ru8>
  namespace: <same namespace as used for infra>
spec:
  config:
    interfaces:
    - ipv4:
```

```
      dhcp: true
      enabled: true
    ipv6:
      enabled: false
    link-aggregation:
      mode: 802.3ad
      options:
        lacp_rate: "1"
        miimon: "140"
        xmit_hash_policy: "1"
      ports:
      - <Bare metal Primary interface's first port/interface name>
      - <Bare metal Primary interface's second port/interface name>
    mac-address: <Bare metal Primary interface's first port's Mac address>
    mtu: <MTU value>
    name: bond0
    state: up
    type: bond
  interfaces:
    - macAddress: <Bare metal Primary interface's first port's Mac address>
      name: <Bare metal Primary interface's first port/interface name>
    - macAddress: <Bare metal Primary interface's second port's Mac address>
      name: <Bare metal Primary interface's second port/interface name>
```

The **NMStateConfig** is dependent on the network environment and must be created in consultation with your network administrators.

Enter the following fields in the NMStateConfig.yaml:
Note: These fields within the **NMStateConfig** are unique per host.

Field	Description
name	Unique name for each node.
namespace	Namespace is the infraenv namespace.
labels	**infraenvs.agent-install.openshift.io: infraNamespace**. The **infraNamespace** must match the name of the **infraenvironment** into which the host is imported.
ports	The correct ports can be found in the **kickstart config** CR. The **kickstart config** for the particular node to be imported to MCE. Within the **kickstart config**, there is a section “NetworkInterfaces”. Within this section, the interfaceName with “baremetal” has the name of the two interfaces interfaceLeg1 and interfaceLeg2. The values under interfaceLeg1 and interfaceLeg2 are the ports. This guidance is for internal. Consult your network administrator for external.
ipaddress	It is the ipv4 address for that node or server.
interfaces > port: macaddress: and port: macaddress:	You can find these values in the **kickstart config** CR. Each port is an entry from the ports. The **macaddress** is for that port. The **macaddress** can be **kickstart config** for the node by looking for the **networkType port: baremetal**. Map that **macaddress** to the **networkCards: -slot-#:** There is one matching **macaddress** and one new one, use those in the **NMStateConfig.yaml**. Important: change any letters to lower case. Example: **ens1f0np1: macaddress:** macaddress can be found in the **kickstart config** for the node under **networkCards: slot-1: macaddress:** This guidance is for internal. Consult your network administrator for external.

NMState information is sourced from the switch. For unmanaged servers, you must manually provide the required values.

After the NMStateConfig.yaml is created for a server, it must be added to IBM Fusion HCI cluster using the following commands:

1. Go to the **infraenvnamespace** namespace.

   ```
   oc project infraenvnamespace
   ```

2. Apply the YAML.

   ```
   oc apply -f nmstate.yaml
   ```

Step 4.2: Download the discovery.iso

Use the **wget** command to download the **discovery.iso** to the service node or another jumpbox.
Note: Keep this ISO secure as you can use this to add a server to the cluster with sensitive information.
To get the retrieval command do the following:

1. Log into the hub IBM Fusion HCI cluster.
2. In the MCE section, go to Infrastructure > Host inventory.
3. In the infrastructure environment, select the infrastructure you intend to import the host into and select that infrastructure environment.
4. Select Add Hosts > With Discovery ISO. There exists either a URL or a **wget** command. Use the command or URL to transfer the ISO to your jumpbox.

Step 4.3: Boot the host with the discovery.iso

This task requires you to log into the IMM, mount the ISO image, and reboot the server.
Note: For internal, go through the steps in this section. For external, go through the Red Hat documentation.
Mount the ISO on to the server through the IMM:

1. Get the IPv6 or IPv4 from kickstart config for that server.
2. Log in to the IMM.
3. To get this value, search for **secretName** field in the **kickstart config**. Inside that secret, search for **defaultUserName** and **defaultUserPasswrd** fields for IMM user and Password respectively.
4. Select Remote Console tab.
5. In the Remote Console tab, select Launch remote console and Media.
6. In the Media page, go to the Mount media from Client Browser section.
7. Select the discovery.iso image and mount it.
8. Select one virtual media to boot on the next restart.
9. Select the mounted discovery.iso and restart immediately.
10. Close windows and monitor that the remote console is showing the reboot.

Step 5: Accept the host into the MCE Inventory

After the server gets restarted with discovery.iso, accept the host into the host inventory.

1. In the MCE user interface, go to Infrastructure > Host inventory.
2. From the table, select the infrastructure environment where the host got imported.
   Note: In the hosts table, you can see the approve host after the server gets booted up completely. It can take 10-15 minutes. Verify the server before you accept. If the server name is not recognizable, then look at the details to confirm. You can change the host name but ensure it is unique.
3. Accept the host into the inventory.

(Optional) Step 6: Labeling hosts

After you accept the hosts, add labels to them through the Agent CR. It can be useful to identify a specific cluster or nodepool.
To add the label, do the following steps:

1. Log into the OpenShift console.
2. Go to Administration > CustomResourceDefinitions.
3. Search for Agent of type `agent-install-openshift.io`.
4. In the instances, find the agent to modify the label.
5. Edit the label of the agent to add a new label.
   Run the following command to label the agent:

   ```
   oc -n <hosted-cluster-namespace> label agent <agent-name> inventory.agent-install.openshift.io/<label-key>=<label-value>
   ```

   For example, If there exists an agent `0176c90f-1111-78cf-1dc6-8d987fe794b2` under namespace `fusion-bm`, then run the following command to apply the label:

   ```
   oc -n fusion-bm label agent 0176c90f-1111-78cf-1dc6-8d987fe794b2 inventory.agent-install.openshift.io/hcp=hcp1
   ```

Deploying Bare Metal clusters with Fusion Data Foundation

You can create a Hosted Control Plane cluster by using Bare Metal hosts either through the OpenShift® Console or the Hosted Control Plane CLI.

Before you begin

- Ensure you meet the following prerequisites:
  - Prerequisites for Hosted Control Plane clusters
  - Planning and prerequisites for your Bare Metal Hosted Control Plane

Procedure

1. Go to Multi Cluster Engine (MCE) user interface.
2. Go to Infrastructure > Clusters > Create cluster.
3. Select Host Inventory and then Hosted.
4. Enter the credentials.
   The credentials should include the following:
   - cloud.openshift.com
   - cp.icr.io
   - quay.io
   - registry.connect.redhat.com
   - registry.redhat.io
5. Enter Cluster name.
   Important: It is crucial that the cluster name is the same name used in the network planning. Do not use a name that does not have an alias entry. The cluster is a part of the base domain. For example, `mydomain.com`. Controller and Infrastructure availability are environment dependent.
   Though the recommendation is to have 3 nodes for a resilient cluster, you can have a single node Hosted Control Plane cluster.
6. Enter Namespace. It is the namespace of the infrastructure environment of the hosts.
7. Enter Labels. These are the labels found on hosts in this particular infrastructure environment. To ensure certain hosts are in a particular cluster, use the labels. Otherwise, a random host from that infrastructure environment gets chosen.
8. Enter the Networking type.
   The value is environment dependent, but you can choose LoadBalancer type for resilience. The SSH public key must be the same that you used during host import.
9. Open the YAML and add the following in the `spec` section: HCP cluster to use the specially created Local Volume Storage for HCP

   ```
   spec:
     etcd:
       managed:
         storage:
           persistentVolume:
             size: 8Gi
             storageClassName: lvms-hcp-etcd
           type: PersistentVolume
     managementType: Managed
   ```

   It causes the Hosted Control Plane cluster to use the Local Volume Storage created for it.

10. Click Create.
    The cluster creation starts and it can take up to half an hour. After the nodepool hosts are in the ready state, add the load balancer for the console to be available.

What to do next

After you deploy the Bare Metal clusters with Fusion Data Foundation, configure the ingress to view the changes in the user interface. For more information about the steps to configure ingress, see Configuring ingress for Bare Metal Hosted cluster.

Configuring ingress for Bare Metal Hosted cluster

After you deploy the Bare Metal clusters with Fusion Data Foundation, you must configure the ingress for you to view the changes in the user interface.

Before you begin

You must have deployed the Bare Metal cluster successfully. For more information about the deployment procedure, see Deploying Bare Metal clusters with Fusion Data Foundation.

Procedure

1. If you choose LoadBalancer networking type, then add a load balancer to the newly created Hosted Control Plane cluster.
   You need the load balancer to gain external access to the Hosted Control Plane cluster. Using a load balancer allows the nodes to be resilient as opposed to the nodeport approach. The load balancer gets added to the newly created Hosted Control Plane cluster, and not the IBM Fusion HCI hub cluster. To access the new Hosted Control Plane cluster, download the kubeconfig. Steps to download the kubeconfig:
   a. Log in to the IBM Fusion HCI hub OpenShift console.
   b. Go to the ACM user interface and select Infrastructre > Clusters.
   c. In the clusters list, select the newly created Hosted Control Plane cluster.
   d. In the Cluster nodepools section, click Download kubeconfig. It downloads the kubeconfig for the cluster.
      After the kubeconfig is available, use the OC commands to create YAMLs on the new Hosted Control Plane cluster.
2. Create metallb operator.
   a. Create a new YAML file.Example YAML:

```
apiVersion: v1
kind: Namespace
metadata:
  name: metallb
  labels:
    openshift.io/cluster-monitoring: "true"
  annotations:
    workload.openshift.io/allowed: management
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: metallb-operator-operatorgroup
  namespace: metallb
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: metallb-operator
  namespace: metallb
spec:
  channel: "stable"
  name: metallb-operator
  source: redhat-operators
  sourceNamespace: openshift-marketplace
```

   b. Apply the YAML:

```
oc --kubeconfig=kubeconfig.yaml apply -f metallb-operator-config.yaml
```

   c. Wait for all the pods to be up and running. Run the following command to confirm:

```
oc --kubeconfig=kubeconfig.yaml get pods -n metallb
```

3. Create an instance of metallb
   a. Create a new file metallb-instance-config.yaml.
      Example:

```
apiVersion: metallb.io/v1beta1
kind:  MetalLB
metadata:
   name: metallb
   namespace:  metallb
```

   b. Apply the file:

```
oc --kubeconfig=kubeconfig.yaml apply -f metallb-instance-config.yaml
```

   c. Wait for the pods to be up and running

```
oc --kubeconfig=kubeconfig.yaml get pods -n metallb
```

4. Create an `IPAddressPool` and `L2Advertisement`.
   a. Create an `IPAddressPool named ipaddresspool-l2advertisement-config.yaml`.
      Example:

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: ippool
  namespace: metallb
spec:
  protocol: layer2
  autoAssign: false
  addresses:
    - 1.23.45.678-1.23.45.678
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: l2-advertisement
  namespace: metallb
spec:
  ipAddressPools:
    - ippool
```

The name of the **IPAddressPool** must match the one in the **L2Advertisement**.

The addresses must be the IPAddress in the DNS that is the ***.apps.NAMEofCluster** in the DNS network table set up by the network administrator.

b. Apply the ipaddresspool-l2advertisement-config.yaml.

```
oc --kubeconfig=kubeconfig.yaml apply -f ipaddresspool-l2advertisement-config.yaml
```

5. Create a service for the **loadbalancer**.
   a. Create metallb-loadbalancer-service.yaml.
   Note: The **address-pool** must match the name of the **addresspool** in previous step.
   Example:

```
kind: Service
apiVersion: v1
metadata:
  annotations:
    metallb.universe.tf/address-pool: ippool
  name: metallb-ingress
  namespace: openshift-ingress
spec:
  ports:
    - name: http
      protocol: TCP
      port: 80
      targetPort: 80
    - name: https
      protocol: TCP
      port: 443
      targetPort: 443
  selector:
    ingresscontroller.operator.openshift.io/deployment-ingresscontroller: default
  type: LoadBalancer
```

   b. Apply the load balancer service YAML.

```
oc --kubeconfig=kubeconfig.yaml apply -f metallb-loadbalancer-service.yaml
```

   c. Monitor the Cluster Operator console for issues:

```
oc --kubeconfig=kubeconfig.yaml get co
```

Expanding your Hosted Control Plane Bare Metal cluster

Add a new host to the exiting Bare Metal Hosted Control Plane cluster.

About this task

After a new Bare Metal host is available in the infrastructure > Nodes > Discovered nodes tab, you can optionally choose to add node to the Hosted Control Plane Bare Metal cluster.

Procedure

1. Create **nmstate** config for new node following **nmstate** YAML template.
   For more information about **nmstate**, see Create the NMStateConfig for each host.
2. Add this new host in the host inventory and boot from the discovery ISO. In MCE, approve the host in the host inventory.
   For more information about these steps, see Download the discovery.iso and Boot the host with the discovery.iso.
   Wait for sometime for the host to be available in MCE. After you approve, it changes to available state. For more information about the steps to accept, see Accept the host into the MCE Inventory.
3. Go to Infrastructure > Clusters > <Your Bare Metal HCP Cluster Name>.
4. Expand Cluster nodepools section and click the ellipsis menu of the node pool.
5. Increase the value of Number of hosts based on the number of new hosts that you plan to add. For example, if you have two nodes in the cluster and want to expand it by one more, then the value of Number of hosts is 3.
6. Click Update.

It initiates the addition of a new node to an existing cluster and completes within a few minutes.

Installing the IBM Fusion base

Procedure to install the IBM Fusion base on a Hosted Control Plane cluster.

About this task

Note: If you plan to use IBM Maximo® in Hosted Control Plane with IBM Fusion, log in to the Hosted Control Plane cluster and deploy Fusion operator.
For the procedure to install IBM Fusion on Bare Metal, see Installing IBM Fusion on On-premises Bare Metal.

- Install Backup & Restore Spoke. See Backup & Restore spoke

Procedure

1. Log in to the OpenShift® Container Platform web console of the IBM Fusion HCI hub, and select All clusters.
2. Go to Infrastructure > Clusters.
   The Clusters page is displayed, and the Cluster list tab is open by default.
3. Go through the list of the associated clusters and their details.
4. You can click the name value link of the cluster to view more information.
   In the cluster details page, the Overview tab is open by default.
5. In the Overview tab page, edit the Labels to include `isf.ibm.com/fusion-base`, and click Save.
   Alternatively, you can edit the label by using the **`oc`** commands:
   a. Log in to the hub cluster.

      ```
      oc login to the hub cluster
      ```

   b. Add label `isf.ibm.com/fusion-base` to the hosted control plan clusters on hub to auto install IBM Fusion consumer cluster:

      ```
      oc label managedcluster <consumer_cluster_name> isf.ibm.com/fusion-base=''
      ```

Installation of data services on a Hosted Control Plane cluster

Provision Hosted Control Plane clusters with services like Fusion Data Foundation, Global Data Platform remote mount, and Backup & Restore.

- **Installing Fusion Data Foundation on a Hosted Control Plane cluster**
  Install the Fusion Data Foundation client as the storage type for the hosted cluster.
- **Deploying storage client in the Hosted Control Plane clusters**
  Steps to deploy storage client in the Hosted Control Plane clusters.
- **Installing Global Data Platform remote mount on the hosted cluster**
  From the hub cluster, install Global Data Platform remote mount on the hosted cluster so that applications can provision storage from a remote-mounted filesystem. It you can use a label-based approach for configuration of IBM Storage Scale CNSA cluster, similar to existing services.

Installing Fusion Data Foundation on a Hosted Control Plane cluster

Install the Fusion Data Foundation client as the storage type for the hosted cluster.

Before you begin

- Ensure that the **`isf.ibm.com/fusion-base`** label is present on the hosted cluster. If the label is not present, install IBM Fusion on the hosted cluster. For more information about the steps to install the IBM Fusion base, see Installing the IBM Fusion base.
- Ensure that prerequisites for Hosted Control Plane clusters are met.

Remember: Fusion Data Foundation is supported on hosted clusters running OpenShift® Container Platform versions N, N-1, N-2, N-3, and N-4, where N is the OpenShift Container Platform version on the hub cluster.

About this task

During the installation of the Fusion Data Foundation client, you can set a storage quota to limit its storage usage.

If the storage quota not assigned during installation, then the storage quota defaults to unlimited for Fusion Data Foundation versions 4.17 and higher.

After the storage quota is set, you cannot reduce it later. Therefore, it is recommended to allocate required storage quota for Fusion Data Foundation clients during the installation of the hosted clusters to ensure the better storage management. For more information about how to set the quota value, see step 4.

Procedure

1. Log in to the OpenShift Container Platform web console and select All clusters.
2. From the Infrastructure menu, select the Clusters tab.
   The Cluster list page appears.
3. In the Cluster list tab, click the ellipsis menu of the required cluster and select Edit labels.

The Edit labels window appears.

4. To allocate a specific storage quota for the Fusion Data Foundation client, do one of the following methods before adding the `isf.ibm.com/fusion-fdf` label:
   Important: If you skip this step and proceed to step 5, the storage quota is set to unlimited by default for this Fusion Data Foundation client (if version is 4.17 or higher). Also, the `isf.ibm.com/fusion-fdf-quota` label appears on the Labels field.
   - **By modifying the Labels field of the hosted cluster:**
     - Add the `isf.ibm.com/fusion-fdf-quota=<value>` label specifying the required storage quota.

       Remember: The storage quota value must match the pattern `<integar-value><Gi/Ti>`, such as 10Gi, 10Ti, or so on. Otherwise, an error message appears on the screen.
       For example:

       ```
       isf.ibm.com/fusion-fdf-quota=300Gi
       ```

       Alternatively, if you can add the `isf.ibm.com/fusion-fdf-quota=<value>` label from the hub cluster by using the following `oc` command:

       ```
       oc label managedcluster <consumer_cluster_name> isf.ibm.com/fusion-fdf-quota='300Gi'
       ```

     - Click Save.
   - **By modifying the configmap of the hub cluster:**
     - Go to the hub cluster.
     - Select the IBM Fusion namespace from the Project list.

       For example, **`ibm-spectrum-fusion-ns`** namespace.

     - Go to Workloads > ConfigMaps.
     - Select the **`fusion-addon-config`** configmap and go to the YAML tab.
     - Add the default-fdf-storage-quota: <value> parameter and click Save.

       Remember: The storage quota value must match the pattern `<integar-value><Gi/Ti>`, such as 10Gi, 10Ti, or so on. Otherwise, an error message appears on the screen.
       For example:

       ```
       default-fdf-storage-quota: 200Gi
       ```

       Important: This storage quota value is set as a default storage quota for all the hosted clusters associated with this hub cluster whenever you add the `isf.ibm.com/fusion-fdf` label. To override it, add the `isf.ibm.com/fusion-fdf-quota=<value>` label with the required value before adding the `isf.ibm.com/fusion-fdf` label.
     - Go back to the All clusters > Infrastructure > Clusters.

       The Cluster list page appears.

     - Click the ellipsis menu of the required cluster and select Edit labels.

       The Edit labels window appears.

5. In the Edit labels window, add the `isf.ibm.com/fusion-fdf` label and click Save.
   Alternatively, if you can add the `isf.ibm.com/fusion-fdf` label from the hub cluster by using the following `oc` command:

   ```
   oc label managedcluster <consumer_cluster_name> isf.ibm.com/fusion-fdf=''
   ```

   The Fusion Data Foundation service starts to install automatically, and it might take 5–10 minutes to complete the installation.
   Note: A notification appears after the successful installation.
6. Verify whether the Fusion Data Foundation client installation is complete and healthy as follows.
   a. Go to the IBM Fusion user interface of the hosted cluster and open the Services tab.
      Ensure that the Fusion Data Foundation service status is healthy.
   b. On the Storage tab, check that the storage status shows as normal and the connection status shows as connected for the local storage.
   c. Go to the hub cluster in the OpenShift Container Platform web console and select the Storage Consumers tab under the Storage menu.
      The Storage Consumers tab displays installed storage consumers with the allotted storage quota.

      To increase the storage quota, see Updating the storage quota of a Hosted Control Plane cluster.

If you face any issues during the installation of the Fusion Data Foundation client, see Troubleshooting issues and known limitations in Hosted Control Plane cluster.

What to do next

To backup applications on the hosted cluster, see Backup & Restore of applications in Hosted Control Plane clusters.

Deploying storage client in the Hosted Control Plane clusters

Steps to deploy storage client in the Hosted Control Plane clusters.

Before you begin

To enable replica -1:

If replica -1 needs to be enabled on the external storage client cluster, ensure that the cluster uses the same topology and that the nodes are labeled in the same way as the hub (provider) nodes. For instructions about enabling replica-1 for external storage clients, see Steps to enable replica -1.

Procedure

Create a storage client in the Hosted Control Plane clusters.

a. Deploy the IBM Fusion in the Hosted Control Plane. For procedure to deploy, see Installing the IBM Fusion base.
b. Install the Data Foundation service and use the following YAML file to create a **FusionServiceInstance** CR. For steps to install Data Foundation, see Installing Fusion Data Foundation and Configuring Fusion Data Foundation in provider mode.

```
apiVersion: service.isf.ibm.com/v1
kind: FusionServiceInstance
metadata:
  name: odfmanager
  namespace: ibm-spectrum-fusion-ns
spec:
  creator: User
  doInstall: true
  parameters:
    - name: namespace
      provided: false
      value: openshift-storage
    - name: creator
      provided: false
      value: Fusion
    - name: backingStorageType
      provided: false
      value: Client
    - name: autoUpgrade
      provided: true
      value: 'false'
    - name: enableLVMStorage
      provided: false
      value: 'false'
  serviceDefinition: data-foundation-service
  triggerUpdate: false
  updateServiceCRSpec: false
```

c. Create **NetworkPolicy** to allow the storage client to connect Data Foundation provider in the host cluster.
Important: This step needs to be done in the host cluster.
   i. Run the following command to get the Hosted Control Plane pods namespace.

```
oc get hostedcontrolplanes.hypershift.openshift.io -A | grep <your-hcp-name>
```

   ii. Create the following YAML and replace the namespace with your Hosted Control Plane cluster. The default value is **clusters-<hcp-cluster-name>**.

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: openshift-storage-egress
  namespace: <your-hcp-namespace>
spec:
  egress:
  - to:
    - namespaceSelector:
        matchLabels:
          kubernetes.io/metadata.name: openshift-storage
  podSelector:
    matchLabels:
      kubevirt.io: virt-launcher
  policyTypes:
  - Egress
```

d. Generate the client onboarding token from the user interface as follows:
Important: The onboarding token is valid for 48 hours and can be used only once.
   i. Log in the OpenShift® Container Platform web console.
   ii. Go to Storage > Storage Clients.
   iii. Click Generating client onboarding token to generate the onboarding token.
   As soon as you click the Generating client onboarding token, the Client onboarding token page is displayed.
   iv. Click Copy to clipboard to copy the token to the clipboard.
e. Get the **storageProviderEndpoint** as follows:
   i. Use the **storageProviderEndpoint** cluster IP.
   ii. Run the following command to get the **storageProviderEndpoint** cluster IP.

```
oc get svc -n openshift-storage ocs-provider-server
```

   Example output:

```
NAME                  TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)     AGE
ocs-provider-server   ClusterIP   172.30.3.161   <none>        50051/TCP   53d
```

f. Create Data Foundation cluster CR as follows:
   i. Create a YAML for Data Foundation cluster CR and replace the **ONBOARDING_TOKEN** and **PROVIDER_SERVICE_ENDPOINT** with the correct value.
   For example:

```
apiVersion: odf.isf.ibm.com/v1
kind: OdfCluster
metadata:
  name: odfcluster
  namespace: ibm-spectrum-fusion-ns
spec:
  storageClient:
    enable: true
    onboardingTicket: $ONBOARDING_TOKEN
    storageProviderEndpoint: '$PROVIDER_SERVICE_ENDPOINT:50051'
  creator: CreatedByFusion
```

   ii. Apply the Data Foundation cluster CR to create the **storageclient**.

g. Verify the `storageclient` status from the IBM Fusion user interface.
h. Verify the Storage Client and StorageClasses details from the OCP user interface.

What to do next

Steps to enable replica -1:

To enable replica -1 for external storage clients, do as follows:

1. Enable the replica -1.

   Command example:

   ```
   oc patch storagecluster ocs-storagecluster -n openshift-storage --type json --patch '[{ "op": "replace", "path": "/spec/managedResources/cephNonResilientPools/enable", "value": true }]'
   ```

2. Add the following annotation to the `StorageConsumer`.

   ```
   failureDomain=$(oc get storagecluster -n openshift-storage -o jsonpath='{.items[0].status.failureDomainKey}' )
   oc annotate storageconsumer <consumer-name> \ -n openshift-storage \ ocs.openshift.io/non-resilient-pools-topology-key="$failureDomain"
   ```

Installing Global Data Platform remote mount on the hosted cluster

From the hub cluster, install Global Data Platform remote mount on the hosted cluster so that applications can provision storage from a remote-mounted filesystem. It you can use a label-based approach for configuration of IBM Storage Scale CNSA cluster, similar to existing services.

Before you begin

- Install IBM Fusion consumer. The label `isf.ibm.com/fusion-base` is required before you add the Global Data Platform remote mount label.
- Meet all the prerequisites for the hosted cluster. See Prerequisites for Hosted Control Plane clusters.

About this task

Note: Global Data Platform remote mount on Hosted Control Plane clusters is supported only on Bare Metal Hosted Control Plane clusters.
You can have a Data Foundation client alongside the CNSA remote service. The Backup & Restore service can be installed with remote-mounted filesystems and it can protect workloads using storage from remote-mounted filesystems.

Procedure

1. Log in to the OpenShift® Container Platform web console of the IBM Fusion consumer cluster.
2. Go to Operators > Installed operators.
3. Check whether the IBM Fusion consumer mode cluster is installed.
   The Status changes from Installing to Succeeded.
4. On the hub cluster, go back to the cluster details > Overview tab page and edit the Labels to include `isf.ibm.com/fusion-gdp-rm`, and click Save.
   Alternatively, you can edit the label from the hub cluster by using the following `oc` command:
   Add label `isf.ibm.com/fusion-gdp-rm` to the hosted control plan clusters on hub to auto install IBM Fusion consumer cluster:

   ```
   oc label managedcluster <consumer_cluster_name> isf.ibm.com/fusion-gdp-rm=''
   ```

5. Go to the IBM Fusion consumer cluster user interface and open the Services page.
   The Global Data Platform remote mount starts to install automatically. If it does not, then refresh and wait for five to ten minutes.
6. Wait for the Global Data Platform remote mount installation to complete.
   A notification is displayed confirming the success of the installation.
7. To backup applications on your hosted plan cluster, see Backup & Restore of applications in Hosted Control Plane clusters.

Backup & Restore of applications in Hosted Control Plane clusters

Using the Backup & Restore service, you can backup and restore your applications on Hosted Control Plane clusters.

Backup and restore your applications on Hosted Control Plane clusters. Here, IBM Fusion hub cluster is where you install the Backup & Restore hub service and Backup & Restore spoke is installed on the hosted cluster. For the steps to backup and restore, see Backup & Restore overview.

- **Installing Backup & Restore on the hosted cluster**
  From the IBM Fusion hub cluster, install Backup & Restore on the hosted cluster with Fusion Data Foundation as the storage type.

Installing Backup & Restore on the hosted cluster

From the IBM Fusion hub cluster, install Backup & Restore on the hosted cluster with Fusion Data Foundation as the storage type.

Before you begin

- Install IBM Fusion consumer and Fusion Data Foundation storage type.
- The label `isf.ibm.com/fusion-base` is required before you add `fusion-backup` label.
- If a specific storageclass is to be used during Backup & Restore installation, then do the following:
  1. Go to the OpenShift® Console on the Hub cluster.
     Note: Changes made in `fusion-addon-config` ConfigMap is applicable to all Hosted Control Planes deployed on the Hub cluster.
  2. In the IBM Fusion namespace, locate the `fusion-addon-config` configmap.
  3. Add an entry:

     ```
     backup-storage-class-name: "NameOfStorageClass"
     ```

     To provide the custom storage class, update the followings fields:

     ```
     data:
       clusters: |
         <cluster-name>: |
           "fusion-install-namespace": "ibm-spectrum-fusion-ns"
           "backup-storage-class-name": "<custom-storageclass-name>"
     ```

     Example YAML:

     ```
     apiVersion: v1
     kind: ConfigMap
     metadata:
       name: fusion-addon-config
       namespace: ibm-spectrum-fusion-ns
     data:
       sds-catalog-source-image: <sds-image>
       storage-quota-min-supported-version: "v4.17"
       clusters: |
         cluster1: |
           "fusion-install-namespace": "ibm-spectrum-fusion-ns"
           "backup-storage-class-name": "ocs-storagecluster-ceph-rbd"
           "backup-restore-spoke-ns": "ibm-backup-restore"
         cluster2: |
           "fusion-install-namespace": "custom-ns"
           "backup-storage-class-name": "ocs-storagecluster-ceph-rbd"
           "backup-restore-spoke-ns": "ibm-backup-restore"
     ```

  Note: If the storage class name is not filled, the Backup & Restore installation makes use of the default storage class on the hosted cluster, prioritizing a storage class with a volume snapshot class. The default values are same as that of the hub cluster values.

Procedure

1. Log in to the OpenShift Container Platform web console of the IBM Fusion hub, and select All clusters.
2. Go to Infrastructure > Clusters.
   The Clusters page is displayed and the Cluster list tab is open by default.
3. Go through the list of the associated clusters and their details.
4. You can click the name value link of the cluster to view more information.
   In the cluster details page, the Overview tab is open by default.
5. In the Overview tab page, edit the Labels to include `isf.ibm.com/fusion-backup`, and click Save.
   Use the default storage class that got created as a part of the Data Foundation for Backup & Restore.
   Use the Backup & Restore to protect your applications but not the entire cluster.

Updating the storage quota of a Hosted Control Plane cluster

Update the allotted storage quota of the Fusion Data Foundation client on a Hosted Control Plane cluster to meet the storage requirement of the application workloads.

Before you begin

Points to remember before starting this procedure:

- You can only increase the storage quota from a defined value to a higher limited value, such as from 90Gi to 100Gi, 100Gi to 200Gi, 9Gi to 1Ti, and so on.
- You can't increase the storage quota from a defined value to unlimited.
- You can't decrease the storage quota.
- You can't modify the storage quota from unlimited to any specific value, as this action is considered a decrease.
- When you set the storage quota, it sets limits on the total storage consumption across the cluster. Therefore, while updating the storage quota, ensure that:
  - The value meets the storage requirements for all application workloads in the cluster.
  - The value meets the storage requirements of all PersistentVolumeClaims (PVCs), regardless of their state within the cluster.

About this task

Use this procedure if you have configured the storage quota for Fusion Data Foundation clients during their installation on hosted clusters and now need to increase it. For more information on storage quota configuration, see Installing Fusion Data Foundation on a Hosted Control Plane cluster.

Whenever the storage consumption of a Fusion Data Foundation client exceeds 80% of its quota limit, you might face issues in creating new applications on the cluster.

You can verify the usage by checking the Storage quota utilization field on the Storage Consumers tab in the OpenShift® Container Platform web console.

If the client's storage consumption exceeds 80% of the quota, a warning icon appears next to the value of the Storage quota utilization field along with the StorageQuotaUtilizationThresholdReached alert. In such cases, increase the client's storage quota to clear the alert.

Note: To see the details of the StorageQuotaUtilizationThresholdReached alert, go to Notifications > Other Alerts.

Procedure

1. Log in to the OpenShift Container Platform web console and select All clusters.
2. Go to Infrastructure > Clusters.
   The Cluster list tab appears on the screen.
3. Click the ellipsis menu of the required cluster and select Edit labels.
   The Edit labels window appears on the screen.
4. Update the `isf.ibm.com/fusion-fdf-quota=<value>` with the required value.
   Remember: The storage quota value must match the pattern `<integar-value><Gi/Ti>`, such as 10Gi, 10Ti, or so on. Otherwise, an error message appears on the screen.
   For example:

   Old storage quota value:

   `isf.ibm.com/fusion-fdf-quota=100Gi`

   New storage quota value:

   `isf.ibm.com/fusion-fdf-quota=1Ti`

5. Click Save.
6. Verify that the Storage Consumers tab displays the updated storage quota for the client as follows.
   a. Go to the hub cluster in the OpenShift Container Platform web console.
   b. Select the Storage Consumers tab under the Storage menu.
   c. Check whether the Storage quota field displays the updated storage quota and the value of the Storage quota utilization field is below 80%.

Results

The Storage Consumers tab displays the updated storage quota for the relevant storage client.

Cleaning up of a Hosted Control Plane cluster

If you no longer need a hosted cluster, use this procedure to remove it.

Before you begin

- Get the cluster ID:
  1. Go to the Red Hat® OpenShift® console of the hosted cluster.
  2. Go to Home > Overview and copy the Cluster ID from the Details section.
- Take a backup of the applications that you want to protect.
- Before you remove the Fusion Data Foundation service, ensure all workloads are stopped: delete the associated PVCs and clean up the storage namespace linked to it.

About this task

If the hosted cluster does not clean up after you remove the base or services label, then give it some time for the changes to take effect. If it still does not clean up, then see Troubleshooting issues and known limitations in Hosted Control Plane cluster.

Procedure

1. On the hub cluster, go back to the hosted cluster details > Overview tab page and edit the Labels to remove `isf.ibm.com/fusion-backup`, and click Save.
2. Remove all the workloads that use Fusion Data Foundation storage classes.
   Important: Fusion Data Foundation cannot be uninstalled if PVCs or VolumeSnapshots exist on the hosted cluster.
3. Edit the Labels again to remove `isf.ibm.com/fusion-fdf`, and click Save.
   Remember: Removing the `isf.ibm.com/fusion-fdf` label also removes the `isf.ibm.com/fusion-fdf-quota=<value>` label, if available.
4. Do the following steps to verify the removal of the Data Foundation storage:
   a. Go to the user interface of the IBM Fusion hosted cluster.
   b. Confirm that the Data Foundation menu option is no longer available.
   c. Go to the Services tab and check whether the Data Foundation service tile is available in the Get started section.
   d. Go to the Red Hat OpenShift console of the hosted cluster and go to Storage > StorageClasses and verify whether all Data Foundation storage classes are removed.
   e. Go to the Storage Clients tab and verify whether the storage client is removed.
      Note: Removing the storage client may take up to 5 minutes.
5. To clean up the Global Data Platform remote mount, complete the following steps:
   a. Delete all applications and PVCs that use storage classes provided by Global Data Platform remote mount.
      Important: Global Data Platform remote mount cannot be uninstalled if applications and PVCs exist on the hosted cluster.
   b. Remove the **`isf.ibm.com/fusion-gdp-rm`** label from the Hosted Control Plane cluster that you want to delete.
6. Delete the hosted cluster.
   a. Log in to the OpenShift Container Platform web console of the IBM Fusion HCI hub, and select All clusters.
   b. Go to Infrastructure > Clusters.
      The Clusters page is displayed and the Cluster list tab is open by default.
   c. Click the name value link of the cluster to view more information.
      In the cluster details page, the Overview tab is open by default.
      Alternatively, you can also select the Destroy clusters from the ellipses overflow menu of the cluster.

   d. In the Actions menu, select Destroy clusters to remove the cluster.

e. In the Permanently destroy clusters window, go through details.
f. Enter the name of the cluster in the Confirm by typing "<cluster name>" below.
g. Click Destroy.
The cluster is removed and its components are deleted. The related namespace is also automatically deleted.

7. To verify the deletion of the hosted cluster, do the following steps:
   a. Log in to the OpenShift Container Platform web console of the IBM Fusion HCI hub, and select All clusters.
   b. Go to Infrastructure > Clusters.
   The Clusters page is displayed and the Cluster list tab is open by default.
   c. Go through the list of the associated clusters and confirm that the Cluster ID of the deleted hosted cluster does not exist.
8. Remove the Backup & Restore connection.
   For the procedure to remove connections, see Disabling the connection.

Troubleshooting issues and known limitations in Hosted Control Plane cluster

Use these troubleshooting information to know the problem and workaround for the installation of spoke cluster and

Hosted Control Plane in Red Hat documentation

For Hosted Control Plane troubleshooting documentation by Red Hat®, see Red Hat Documentation.

PVCs on Hosted Control Plane clusters are stuck in terminating state

If PVCs on the Hosted Control Plane clusters are stuck in terminating state, then do the following steps for all applications that have PVCs associated to Fusion Data Foundation storage.

1. Scale down deployment of the PVC application to 0.
2. Delete application.
3. Check the PVC status to confirm whether it is deleted.
   Note: Remove all PVCs from the applications before you remove the Fusion Data Foundation label.
4. Check whether the Fusion Data Foundation client on the Hosted Control Plane cluster is deleted.

Issues in hosted cluster clean up

Issues in Fusion base cleanup

If the hosted cluster does not clean up after you remove the base label (`isf.ibm.com/fusion-base`), then give it some time for the changes to take effect. If it still does not clean up, follow these steps to understand the cause of the issue:

1. Log in to the hosted cluster.
2. Go to pods in the `open-cluster-management-agent-addon` namespace.
3. Get the terminal access to the `fusion-cleanup-agent` pod.
4. Run the following **cleanup** script:

   ```
   bash /scripts/sds/sds-cleanup.sh
   ```

5. If the script stuck at any resource deletion, remove the finalizer from that resource.

Issues in Fusion Data Foundation service clean up

Before you begin:

- Before you proceed with Fusion Data Foundation deletion, stop the backup service and clean up the associated PVCs.
- If Fusion Data Foundation sizes are large and workloads are running continuously, such as Cloud Pak or Watsonx, then do the following steps:
  1. Scale the pods down to zero.
  2. Stop the application.
  3. Delete the associated PVCs.

If the Fusion Data Foundation service does not clean up after you remove its label (`isf.ibm.com/fusion-fdf`), then give it some time for the changes to take effect. If it still does not clean up, follow these steps to understand the cause of the issue:

1. Log in to the hosted cluster.
2. Go to pods in the `open-cluster-management-agent-addon` namespace.
3. Get the terminal access to the `fusion-odf-cleanup-agent` pod.
4. Run the following **cleanup** script:

   ```
   bash /scripts/odf/delete-fusion-odf.sh
   ```

5. If the script stuck at any resource deletion, remove the finalizer from that resource.

Issues in Backup & Restore service clean up

If the Backup & Restore service does not clean up after you remove its label (`isf.ibm.com/fusion-backup`), then give it some time for the changes to take effect. If it still does not clean up, follow these steps to understand the cause of the issue:

1. Log in to the hosted cluster.
2. Go to pods in the `open-cluster-management-agent-addon` namespace.
3. Get the terminal access to the `fusion-bnr-cleanup-agent` pod.
4. Run the following **cleanup** script:

   ```
   bash /scripts/backup-restore/uninstall-backup-restore.sh
   ```

5. If the script stuck at any resource deletion, remove the finalizer from that resource.

Pull image issue occurs during the installation of IBM Fusion in a hosted cluster

Problem statement

If a pull image issue occurs, ensure that the pull-secret for `cp.icr.io` is available to the Hosted Control Plane cluster.

Resolution

To check and update the pull-secret, do the following steps:

1. On the hub cluster, go to Administration > CustomResourceDefinition and search for `HostedCluster`.
2. Look for the instance that corresponds with the Hosted Control Plane cluster and open it in YAML view.
3. Search for `pullSecret` and note down the name of the secret.
4. In the clusters namespace, search for that secret and open it in edit mode.
5. Check to see whether the correct value of `cp.icr.io` available. If not, modify the secret with the right value.
   It can take a while to propagate. You may even have to approve the changes from the Compute > Nodes page of the Hosted Control Plane cluster.

It is recommended to install a Hosted Control Plane cluster with the following credentials:

- `cloud.openshift.com`
- `cp.icr.io`
- `quay.io`
- `registry.connect.redhat.com`
- `registry.redhat.io`

Issues in installation of IBM Fusion in a hosted cluster

Resolution

1. Check the status of `manifestwork` `fusion-install` in `clusternamespace` of the hub cluster:

```
oc login to the hub
oc get manifestwork fusion-install -n <spoke_cluster_name> -o yaml
```

2. If no error exists in the status of the `manifestwork`, check the status of IBM Fusion installation in the spoke cluster:

```
oc login to the spoke
oc get csv -n ibm-spectrum-fusion-ns
```

Issues in the installation of Fusion Data Foundation in the hosted cluster

Resolution

As a resolution, check the status of `manifestwork` `odfclient-install` in `clusternamespace` in the hub cluster:

```
oc login to the hub
oc get manifestwork odfclient-install -n <spoke_cluster_name> -o yaml
```

Backup issues in Hosted Control Plane with Fusion Data Foundation

Problem statement

Concurrent backups fail during a Velero snapshot with the following error message:

```
The  operation has timed out because Velero has failed to report status.'
     However, the backup phase is updated as 'DataTransferfailed.
```

The timeout for Velero to pickup request is set to 30 minutes in the transaction manager.

Resolution

To resolve the issue, increase it to 60 minutes.

Hosted Control Plane cluster does not get created because of unavailability of IP addresses

Resolution

Check whether the installed load balancer has sufficient IP addresses for the Hosted Control Plane cluster. Check the `IPAddressPool` object for the IP range. Run the following command to check whether an IP is available:

```
oc get svc -A | grep LoadBalancer
```

Known issues and limitations

- During the installation or upgrade of the Hosted Control Plane, you might encounter an issue where the `control-plane-operator` pods restart continuously and increase in number in the `clusters-<hcp_name>` namespaces on the host cluster.
- Random "image pull" failure can occur on the Hosted Control Plane due to secrets.
- Sometimes, the hosted cluster status goes in offline mode after deletion. Contact IBM Support to resolve the issue.
- The **hcp destroy** command can cause the hosted cluster to get stuck indefinitely during cleanup. Contact IBM Support to resolve the issue.
- The YAML tab view on the OpenShift® Container Platform console is not working as expected during the installation of the multi cluster engine operator.
- The hosted clusters expect the `storageprofile` of the used storage class, but it is not available during cluster creation.
- The following issues occur whenever you remove the disks and place them back in the rack:
  - Disks are not reflected in the node.
  - LVM cluster and Data Foundation go into a degraded state.

As a resolution, restart the affected node.

Red Hat OpenShift Virtualization

IBM Fusion enables the coexistence of containers and VMs, with OpenShift Virtualization allowing VMs to be managed as Kubernetes resources. Beyond VM hosting, Fusion and OpenShift Virtualization offer virtual machine mobility, backup and restore, and additional capabilities to ensure high availability and disaster recovery.

- **Overview**
  The Red Hat OpenShift provides a platform for running virtual machines and container workloads side by side. It relieves you of maintaining separate infrastructure, platforms, tools, skills, and teams for virtual machines and containers. It also enables virtual machines to be managed with the same modern practices that are used for containers.
- **Red Hat OpenShift Virtualization Sizing on IBM Fusion**
  Sizing in Red Hat OpenShift Virtualization is to arrive at the optimal configuration of the infrastructure resources to efficiently accommodate the number and types of virtual machines (VMs) you plan to deploy.
- **Setting storage for virtual machines**
  Virtual machines use block storage and file storage.
- **Deploying Red Hat OpenShift Virtualization Operator**
  Procedure to deploy the Red Hat OpenShift Virtualization Operator.
- **Migrating VMs to OpenShift Virtualization**
  To move from traditional VM-based infrastructure to a Kubernetes-native environment, migrate virtual machines (VMs) to OpenShift Virtualization (OSV), which integrates VMs and containers within the same platform.
- **Backup and restore of virtual machines**
  IBM Fusion protects virtual machines from data loss and corruption by backing up the virtual machines regularly. The virtual machines can then be restored from a previous backup.
- **Creating a virtual machine by using Clone PVC**
  A clone is an exact duplicate of an existing storage volume that can be used like any standard volume. You create a clone to capture a point-in-time snapshot of the data.
- **Disaster recovery for virtual machines**
  Disaster Recovery of virtual machines involves creating regular backups, maintaining up-to-date replicas, and implementing robust failover mechanisms. Implementing failover mechanisms ensures high availability and minimizes potential data loss, reducing downtime, and ensuring long-term resilience against disasters.

Overview

The Red Hat® OpenShift® provides a platform for running virtual machines and container workloads side by side. It relieves you of maintaining separate infrastructure, platforms, tools, skills, and teams for virtual machines and containers. It also enables virtual machines to be managed with the same modern practices that are used for containers.

IBM Fusion provides storage, encryption at rest, and resiliency to run virtual machines on Red Hat OpenShift.

The RWX (ReadWriteMany) mode is crucial because it enables Live Migration for virtual machines in Red Hat OpenShift. With RWX block storage support, virtual machines can seamlessly move between nodes within the OpenShift cluster during Live Migration while maintaining the performance benefits of the block storage.

Virtual machine resiliency can be provided in the following ways:

- High availability is achieved by using three-way replication to mirror data across availability zones. It allows workloads to continue to run even when storage infrastructure fails.
- RWX storage enables Live Migration, which allows virtual machines to be moved between nodes in the cluster.
- Backup capabilities protect virtual machines from data loss or corruption, enabling them to be restored to an earlier point in time. You can restore the virtual machines to an alternative cluster to recover from a complete cluster failure or practice recovery.
- When site outage occurs, use Metro-DR or Regional disaster recovery to recover virtual machines.

Red Hat OpenShift Virtualization Sizing on IBM Fusion

Sizing in Red Hat® OpenShift® Virtualization is to arrive at the optimal configuration of the infrastructure resources to efficiently accommodate the number and types of virtual machines (VMs) you plan to deploy.

With proper sizing, your infrastructure can efficiently manage the expected workload, allowing you to allocate the right resources to optimize both performance and cost.

Sizing considerations

Some of the factors that influence the sizing of virtual machines on IBM Fusion are as follows:

CPU
: The number of vCPUs depend on the computational demands of your VMs. VMs that run resource-intensive applications or multiple workloads require more vCPUs to perform efficiently.

Memory
: Memory allocation is crucial for VMs to run smoothly and must be sized according to the specific workload requirements of each VM. Applications that perform data-intensive operations or handle large in-memory datasets require more memory to maintain optimal performance.

Storage

The storage depends on the volume of data that each VM is expected to handle. It includes the storage for operating systems, application data, and any additional storage for backups or high-availability configurations.

Networking
: While not directly affecting the number of VMs, network bandwidth and latency are crucial for the performance of VMs, especially in environments with high data transfer rates or low-latency requirements.

Workload Characteristics
: The type of applications and their resource usage patterns significantly impact the sizing considerations. For example, CPU-bound, memory-bound, I/O-bound.

Redundancy and High Availability
: Configurations for redundancy and high availability (HA) may require additional resources, such as extra storage or memory, to help ensure that VMs can failover without service interruption.

Extra capacity considerations

When you plan your sizing, consider the capacities that are needed for CoreOS, OpenShift Container Platform, and IBM Fusion to ensure that your VMs have access to the necessary resources.

Go through the following resource requirements for the various components:

IBM Fusion software
: IBM Fusion Software System Requirements

OpenShift Container Platform
: OpenShift Container Platform Resource Requirements

Data Foundation
: Planning Fusion Data Foundation deployment

Before your allocate resources to VMs, subtract these overhead values from your total available resources.

- **Sizing calculator**
  Use the following formulas to provide an estimated infrastructure for VM deployments. Considering the overhead and high availability, these estimates offer a close approximation for effective resource planning.

Sizing calculator

Use the following formulas to provide an estimated infrastructure for VM deployments. Considering the overhead and high availability, these estimates offer a close approximation for effective resource planning.

Expected input

Number of VMs
: Total number of virtual machines planned.

VM Types
: Are the VMs of the same type or different types with distinct vCPU, Memory, and Storage requirements?

Formulas

- **CPU Capacity:** The following formula calculates the total vCPU capacity that is available for VMs. Subtract the overhead for control plane and other necessary services:
  Total Available (vCPUs) = (vCPUs_per_node × node_count − control_plane_requirements − overhead) × (1−ha_reserve_percent) × (overcommit_ratio)

Parameter	Description
vCPUs_per_node	The number of vCPUs per compute node. For example, with 64 cores and 2 threads per core, you would have 128 vCPUs per node.
node_count	The number of compute nodes.
control_plane_requirements	CPU is reserved for control plane functions.
overhead	Overhead for management services. For example, Data Foundation in IBM Fusion.
ha_reserve_percent	Percentage of capacity reserved for high availability.
overcommit_ratio	Factor to consider when you overcommit CPU resources. For example, 4:1.

- **Memory Capacity:** Similarly, the memory capacity calculation accounts for the overhead and reserves for high availability:

  Total Available Memory (GiB) = (total_node_memory × node_count − control_plane_requirements − overhead) × (1 − ha_reserve_percent)

Parameter	Description
total_node_memory	Total physical memory per compute node.
control_plane_requirements	Memory reserved for control plane operations.
overhead	Overhead for management services. For example, Data Foundation in IBM Fusion.
node_count	Number of compute nodes.
ha_reserve_percent	Percentage of memory reserved for high availability.

- **Storage Capacity:** Storage capacity is calculated after accounting for the overhead needed for management and other services:
  Total Available Storage (TiB) = total_storage_capacity − storage_overhead

Parameter	Description
total_storage_capacity	Total storage capacity available.
storage_overhead	Overhead for management services and storage redundancy.

Example

Table with example VM specification:

VM specification	Description
50 VMs of Type 1	Each VM requires 2 vCPUs, 8 GiB of memory, and 100 GiB of storage.
20 VMs of Type 2	Each VM requires 4 vCPUs, 16 GiB of memory, and 200 GiB of storage.

Table with example infrastructure Resources (Before Overhead and HA)

Infrastructure Resources (Before Overhead and HA)	Description
Compute Nodes	13 nodes, each with 64 cores and 2 TB (2048 GiB) of memory.
Total Raw Storage	16 nodes (13 compute + 3 control), each with 10 disks of 7.68 TiB each.

Table with example overhead and high availability reserves

Overhead and high availability reserves	
CPU: Overhead	32x4 = 128 (assuming each core provides 4 vCPUs) reserved for management services.
CPU: High Availability (HA) Reserve	20% of the remaining vCPUs.
CPU: Control Plane Requirements	4x4= 16 vCPUs (assuming each core provide 4 vCPUs) reserved for control plane operations.
Memory: Overhead	72 GiB reserved for management services.
Memory: High Availability (HA) Reserve	20% of the remaining memory.
Memory: Control Plane Requirements	24 GiB reserved for control plane operations.
Storage: Total Raw Storage	16×10×7.68 TiB = **1228.8 TiB**
Storage: Usable Storage After 3-Way Replication	(1228.8)/3 TiB = **409.6 TiB**
Storage: Storage Overhead	Assume 10% overhead for management services and system needs: **40.96 TiB**
Storage: Total Available Storage After Overhead	Total Available Storage= (Total Usable Storage) - (Storage Overhead)= 409.6 TiB - 40.96 TiB= **368.64 TiB**

Table showing CPU calculation

CPU	Value
Total Physical CPU Cores	Total Physical CPU Cores= 64×13= 832 cores
Total vCPUs	256x13 = 3328 vCPUs (assuming each core provides 4 vCPUs, per node would be 64x4 = 256)
Control Plane Requirements	4x4= 16 vCPUs (assuming each core provide 4 vCPUs) are reserved for control plane operations.
Overhead	32x4 = 128 (assuming each core provides 4 vCPUs) are reserved for management services.
High Availability (HA) Reserve	20% of the remaining vCPUs.
Total Available vCPUs After Control Plane, Overhead, and HA Reserve	Total Available (vCPUs) = (vCPUs_per_node × node_count – control_plane_requirements – overhead) × (1–ha_reserve_percent) × (overcommit_ratio)
Substituting the values	Total Available vCPUs = (3328 – 16 – 128) × 0.8 × 4 = 10188.8 vCPUs (Overcommit ratio is assumed to be 4:1) Rounding it down to 10189 vCPUs available for VM deployments.
Comparison with Available CPU	180 vCPUs required < 10189 vCPUs available. Rounding it down to 10189 vCPUs available for VM deployments.

Result: The infrastructure has sufficient CPU resources to support the required VMs even after considering the control plane requirements.

Table showing memory calculation

Memory	Values
Total Physical Memory	Total Physical Memory=2048×13=26624 GiB
Control Plane Requirements	24 GiB reserved for control plane operations.
Overhead	72 GiB reserved for management services.
High Availability (HA) Reserve	20% of the remaining memory.
Total Available Memory After Control Plane, Overhead, and HA Reserve	*Total Available Memory (GiB) = (total_node_memory × node_count – control_plane_requirements – overhead) × (1 – ha_reserve_percent)*
Substituting the values	Total Available Memory (GiB)=(2048×13 – 24 – 72) × 0.8= 26528×0.8= 21222.4 GiB Rounding it down to **21222 GiB** available for VM deployments.
Comparison with Available Memory	720 GiB required < 21222 GiB available

Result: The infrastructure has sufficient memory resources to support the required virtual machines even after considering the control plane requirements, overhead, and HA.

Table showing storage calculation

Storage	Values
Total Raw Storage	Total Raw Storage=16×10×7.68 TiB= 1228.8 TiB
Total Usable Storage After 3-Way Replication	Total Usable Storage=(1228.8)/3 TiB= 409.6 TiB
Storage Overhead	Storage Overhead= 40.96 TiB
Total Available Storage After Overhead:	Total Available Storage=409.6 TiB – 40.96 TiB= 368.64 TiB
Total Storage required for Type 1 VMs:	Storage for Type 1 VMs= 50×100 GiB= 5000 GiB= 5 TiB
Total Storage required for Type 2 VMs:	Storage for Type 2 VMs= 20×200 GiB= 4000 GiB= 4 TiB
Total Storage required for all VMs:	Total Storage required= 5 TiB+4 TiB= 9 TiB
Comparison with Available Storage:	9 TiB required < 368.64 TiB available

Conclusion

- CPU: 10189 vCPUs available; 180 vCPUs required (Sufficient)
- Memory: 21222 GiB available; 720 GiB required (Sufficient)
- Storage: 368.64 TiB available; 9 TiB required (Sufficient)

The infrastructure can support the deployment of 50 Type 1 VMs and 20 Type 2 VMs, with sufficient CPU, memory, and storage resources. These resources are available after accounting for overhead and high availability reserves.

Setting storage for virtual machines

Virtual machines use block storage and file storage.

Block-mode storage through the **ocs-storagecluster-ceph-rbd-virtualization** storage class is recommended as it provides optimized performance for block IO.

However, for Windows VMs, it is always recommended to use **ocs-storagecluster-ceph-rbd-virtualization** storage class. If you use the **ocs-storagecluster-ceph-rbd** storage class, then Windows VMs may report bad CRC or signature errors due to I/O patterns. The performance of the cluster performance can degrade due to excessive re-transmissions caused by CRC errors that leads to network saturation.

VMs must request RWX storage so that they can be moved between nodes non-disruptively using Live Migration.

- File storage is supported for virtual machines, but it lacks the performance advantage of block storage.
- RWO volumes are supported for virtual machines but are not recommended, as they block Live Migration. VMs using RWO volumes cannot move between nodes without causing disruption.

For more information about live migration, see Live migration for virtual machines.

Fusion Data Foundation can be configured for encryption, with keys managed within the OpenShift cluster or by an external Key Management Service (KMS). For more information about encryption, see Preparing to connect to an external KMS server in Fusion Data Foundation.

Deploying Red Hat OpenShift Virtualization Operator

Procedure to deploy the Red Hat® OpenShift® Virtualization Operator.

About this task

OpenShift Virtualization provides a platform to bring virtual machines (VMs) into containerized workflows by running a virtual machine within a container where you develop, manage, and deploy virtual machines side-by-side with containers in one platform. For more information about how to create and manage VMs, see Red Hat OpenShift Virtualization and What you can do with OpenShift Virtualization.

A list of example VM features that the Red Hat OpenShift Virtualization supports:

- Create and manage Linux and Windows virtual machines
- Connect to virtual machines through a variety of consoles and CLI tools
- Import and clone existing virtual machines
- Manage network interface controllers and storage disks attached to virtual machines

Procedure

1. Go to the Red Hat OpenShift Container Platform Operator Hub.
2. Search for OpenShift Virtualization.
3. Install the OpenShift Virtualization Operator.
4. Configure the installation parameters to either custom or recommended namespace.
5. Wait for the installation to complete.
6. Go to OpenShift Virtualization in Installed Operators page and click Create Hyperconverged to configure the Hyperconverged resource.
   Note: When you create the **HyperConverged** CR, ensure you configure the appropriate eviction strategy for your virtual machines in the **HyperConverged** CR. For more information, refer to the Eviction strategy documentation.
7. When the Operator deployment is complete go to Installed Operators and look for OpenShift Virtualization within the **openshift-cnv** or your namespace.
8. Check whether the status is successful.
   A new Virtualization menu appears. It indicates that your cluster is now ready to deploy virtual machines within a project /or namespace of your choice.

What to do next

- You can create virtual machines with any default size or custom sizes.
- Use **ocs-storagecluster-ceph-rbd-virtualization** storage class for the OS disk of VMs. For more information about setting storage, see Setting storage for virtual machines.

- **Configuring GPU access with NVIDIA vGPU**
  NVIDIA vGPU technology enables a single physical GPU to be divided into multiple virtual instances, allowing multiple virtual machines to share GPU resources. This improves GPU utilization, reduces infrastructure costs, and maintains performance isolation for AI, data analytics, and graphics workloads.

Configuring GPU access with NVIDIA vGPU

NVIDIA vGPU technology enables a single physical GPU to be divided into multiple virtual instances, allowing multiple virtual machines to share GPU resources. This improves GPU utilization, reduces infrastructure costs, and maintains performance isolation for AI, data analytics, and graphics workloads.

Important: Ensure that your GPU hardware is supported by NVIDIA vGPU technology. To get the complete list of supported GPU cards, see NVIDIA documentation.
For the procedure to configure NVIDIA vGPU for virtual machines in OpenShift® Virtualization, see NVIDIA documentation.
Important: While you follow the NVIDIA documentation, take note of the following two steps that must be done differently:

1. To enable sandbox workloads for vGPU, you must configure additional parameters in the **ClusterPolicy** for the GPU operator.

```
sandboxWorkloads.enabled=true
sandboxDevicePlugin.enabled=true
sandboxWorkloads.defaultWorkload=vm-vgpu
vfioManager.enabled=false
```

2. After completing the step creating a **ClusterPolicy** for the GPU operator in the NVIDIA documentation, you must label the GPU node with the specific vGPU device profile.

```
oc label node <gpu-node-name> \
  nvidia.com/vgpu.config=<vgpu-device-name> \
  --overwrite
```

Replace **<gpu-node-name>** with the name of your GPU-enabled node, and **<vgpu-device-name>** with the vGPU profile that you want to enable. For example, on NVIDIA RTX Pro 6000 Blackwell GPUs, a valid vGPU profile is **DC-48Q**. You can find the appropriate **vgpu-device-name** in the **default-vgpu-devices-config ConfigMap**, or in a custom **ConfigMap** if created.

To identify the correct profile, search for the **device_id** that matches your GPU, using the last four characters of the device ID as a reference.

Migrating VMs to OpenShift Virtualization

To move from traditional VM-based infrastructure to a Kubernetes-native environment, migrate virtual machines (VMs) to OpenShift® Virtualization (OSV), which integrates VMs and containers within the same platform.

- **Migrating VMs from VMware vSphere to OpenShift**
  OpenShift Virtualization supports the migration of VMs from VMware vSphere to OpenShift.

Migrating VMs from VMware vSphere to OpenShift

OpenShift® Virtualization supports the migration of VMs from VMware vSphere to OpenShift.

Before you begin

- Ensure that the internal OpenShift registry setup is complete. Use an OpenShift internal registry image from a single namespace to provision workloads across multiple namespaces. After the internal OpenShift registry setup is complete, follow this procedure to use an OpenShift internal registry image that is stored in a single namespace to provision workloads across desired namespaces.
- Deploy the Migration Toolkit for Virtualization (MTV) operator. For more information about deploying MTV operator, see Installing the Migration Toolkit for Virtualization.

About this task

OpenShift Virtualization supports the migration of VMs from VMware vSphere to OpenShift. To accelerate this process, it uses Virtual Disk Development Kit (VDDK) images. VDDK enables faster, block-level access to VM disks, reducing transfer times by skipping unused data and improving overall performance. It is useful for large VMs or bulk migrations. For more information about how to build and configure the VDDK image, see OpenShift.

Note: In this procedure, Origin Namespace refers to where the internal image is stored and Destination Namespace refers to where the image are pulled and used by pods.

Procedure

1. Create a Service Account in the destination namespace.
   It is the namespace where your pods run and provides access to the image.

```
oc create serviceaccount registry-puller -n <destination-namespace>
```

2. Create an Image Pull Secret in the destination namespace.
   Go through the internal OpenShift registry and provide your credentials.

```
oc create secret docker-registry registry-pull-secret \
  --docker-server=<internal-openshift-image-registry> \
  --docker-username=<your_username> \
  --docker-password=<your_password> \
  --docker-email=<your_email> \
  -n <destination-namespace>
```

3. Link the Secret to the Service Account in the destination namespace.

```
oc secrets link registry-puller registry-pull-secret --for=pull -n <destination-namespace>
```

4. Grant Permission to the Service Account to Pull from the origin namespace.
   It allows the service account in the destination namespace to pull images from the origin namespace.

```
oc adm policy add-role-to-user system:image-puller \
  system:serviceaccount:<destination-namespace>:registry-puller \
  -n <origin-namespace>
```

High-availability of virtual machines

High availability of virtual machines (VMs) ensure critical applications and services are available even in the event of unplanned scenarios.

IBM Fusion supports the following features for the high availability of VMs.

Live Migration
: Use live migration to move VMs from one host to another without downtime. This is useful for load balancing and maintenance. See Live migration for virtual machines.

Failover Clustering
: Configure VMs in a failover cluster, which allows VMs to automatically restart on a different host if the original host fails. It is very useful in the event of any unexpected hardware failures or other unplanned incidents. See Virtual machine failover.

Live migration for virtual machines

High availability of virtual machines can be achieved by live migration as the virtual machine continues to be operative and accessible. Live migration is the process of moving a virtual machine that is running from one physical host to another seamlessly without interrupting the workload.

Before you begin

In the `openshift-cnv` namespace, edit the `HyperConverged` CR and add the necessary live migration parameters:

```
oc edit hyperconverged kubevirt-hyperconverged -n openshift-cnv
```

Example:

```
apiVersion: hco.kubevirt.io/v1beta1
kind: HyperConverged
metadata:
  name: kubevirt-hyperconverged
  namespace: openshift-cnv
spec:
  liveMigrationConfig:
    bandwidthPerMigration: 64Mi
    completionTimeoutPerGiB: 800
    parallelMigrationsPerCluster: 5
    parallelOutboundMigrationsPerNode: 2
    progressTimeout: 150
```

About this task

Manually initiate a live migration of a virtual machine instance to another node using either the web console or the CLI. For more information about live migration, see Red Hat documentation.

Live migration from the user interface

Procedure

1. As an administrator user, log in to the OpenShift® Container Platform console.
2. Go to Virtualization > VirtualMachines.
3. Search for the virtual machine that you want to migrate.
4. Click the ellipses overflow menu of the virtual machine and select Migrate or you can go to the virtual machine Overview screen and select Migrate Virtual Machine from the Actions menu.
   The status of the virtual machine changes from Running to Migrating.
5. After the migration completes, in the virtual machine details page, check the hosting node name to confirm the migration.

Live migration from the command line

Procedure

1. Create a migrate.yaml configuration file for the virtual machine instance to migrate.
   migrate.yaml:

```
apiVersion: kubevirt.io/v1
kind: VirtualMachineInstanceMigration
metadata:
  name: {{ vm_name }}-migrate
  namespace: {{ namespace }}
spec:
  vmiName: {{ vm_name }}
```

Example:

```
apiVersion: kubevirt.io/v1
kind: VirtualMachineInstanceMigration
metadata:
  name: migration-job
  namespace: openshift-cnv
spec:
  vmiName: icsvm-storage-gui-kocmddajec
```

2. Apply the YAML file to initiate the migration.

```
oc apply migrate.yml
```

Virtual machine failover

In IBM Fusion, you can achieve automatic failover during unplanned node failures by using MachineHealthCheck and Self Node Remediation operators.

About this task

A node can be shutdown in a planned graceful way or unexpectedly because of reasons such as power outage or other external factors. A node shutdown could lead to workload failure if the node is not drained before the shutdown. A node shutdown can be either graceful or non-graceful.

Without VM failover automation, a worker node failure leaves the VMs on that node in a state of uncertainty, waiting for recovery. This downtime can cause the application to become unavailable until the node is back online. After you set up the failover mentioned in this procedure, the VMs running on the failed node are automatically migrated to another available healthy node. It ensures high availability and minimizes downtime without user intervention. It does not cover live migration for maintenance or manual node migrations. For more information about live migration in , see Live migration for virtual machines.

In the event of a non-graceful node shutdown, you can also manually initiate a failover by stopping the `kubelet` service on the node.

```
sudo systemctl stop kubelet.service && sleep 360s && sudo systemctl start kubelet.service
```

Important: Do not use the examples that are mentioned in this procedure as it is; instead, replace it with values relevant to you.

Procedure

1. Check whether the Virtualization operator is in a healthy state on the cluster.
2. Install the Self Node Remediation (SNR) operator by Red Hat® on the cluster.
   Self Node Remediation Operator (SNR) is used as remediation operators to fence and recover the affected nodes and workloads.The node health check operator automatically installs this operator. If it is not installed, go to Operator hub and install the Self Node Remediation Operator.

   For more information about self node remediation, see Red Hat Documentation.

   To deploy in an offline environment, mirror these operator images in the Red Hat operator catalog index image.

3. Update the `SelfNodeRemediationConfig` Custom Resource (CR) and create the `SelfNodeRemediationTemplate` (CR) with strategy as `OutOfServiceTaint`.
   By default, when Self Node Remediation operator is installed, it creates the default remediation template that uses the automatic strategy.
   Note: Update this strategy to `OutOfServiceTaint`. You want the nodes to be tainted to out of service if any node goes down.
   Do the following steps to update the Self Node Remediation Operator strategy:
   a. In the Operators > Installed Operators, search and open the Self Node Remediation Operator.
   b. Go to the Self Node Remediation Template tab.
   c. Click Create SelfNodeRemediationTemplate to create the CR of SNR operator in the `openshift-machine-api` namespace.
      By default, the `SelfNodeRemediationTemplate` is created in the `openshift-workload-availability` namespace. Create the template within the `openshift-machine-api`.
   d. Update `spec.template.spec.remediationStrategy` to `OutOfServiceTaint`.
      Create a template under `openshift-machine-api` namespace similar to the following example:

```
apiVersion: self-node-remediation.medik8s.io/v1alpha1
kind: SelfNodeRemediationTemplate
metadata:
  annotations:
    remediation.medik8s.io/multiple-templates-support: 'true'
  name: self-node-remediation-oost-strategy-template
  namespace: openshift-machine-api
  labels:
    remediation.medik8s.io/default-template: 'true'
spec:
  template:
    spec:
      remediationStrategy: OutOfServiceTaint
```

   e. Go to the Self Node Remediation Config tab.
   f. Open the self node remediation config file to update it.
   g. Open the YAML tab.
      When the SNR is installed, it creates the `SelfNodeRemediationConfig` CR that contains the configuration information.
      Example:

```
apiVersion: self-node-remediation.medik8s.io/v1alpha1
kind: SelfNodeRemediationConfig
metadata:
  generation: 2
name: self-node-remediation-config
  namespace: openshift-workload-availability
  uid: b0c8e0b7-e433-47e3-94d2-e5ebf98da085
spec:
  apiServerTimeout: 5s
  peerApiServerTimeout: 5s
  hostPort: 30001
  isSoftwareRebootEnabled: false
  watchdogFilePath: /dev/watchdog
  peerDialTimeout: 5s
  peerUpdateInterval: 15m
  apiCheckInterval: 15s
```

```
  peerRequestTimeout: 5s
  maxApiErrorThreshold: 3
```

Note: By default, when the SNR is installed, it creates the **SelfNodeRemediationConfigCR**. For more information about the configuration, see Red Hat documentation. Update the **spec.isSoftwareRebootEnabled** state to `false` so that the node does not reboot. During node failure, the node must be tainted to prevent scheduling of new pods without rebooting.

4. From the OpenShift menu, go to Compute > MachineHealthChecks and create a new Machine Health Check (MHC) under your namespace.
   The Machine Health Check (MHC) checks the health of nodes and triggers the remediation agents if the node status is Unknown or NotReady for a defined amount of time. For more information about the procedure, see Red Hat Documentation.
5. Open the MachineHealthChecks file details and click YAML.
   Example MachineHealthCheck CR:

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineHealthCheck
metadata:
  name: machinehealthcheck
  namespace: openshift-machine-api
spec:
  maxUnhealthy: 40% # Allow up to 40% of worker nodes to be unhealthy at once before remediation
  nodeStartupTimeout: 20m #node does not join the cluster after the nodeStartupTimeout, the machine is remediated.
  remediationTemplate:
    apiVersion: self-node-remediation.medik8s.io/v1alpha1
    kind: SelfNodeRemediationTemplate
    name: self-node-remediation-oost-strategy-template
    namespace: openshift-machine-api
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <cluster-machineset-name>
      machine.openshift.io/cluster-api-machine-role: <role>
      machine.openshift.io/cluster-api-machine-type: <role>
      machine.openshift.io/cluster-api-machineset: <cluster_name>-<label>-<zone>
  unhealthyConditions:
    - status: Unknown
      timeout: 10s # Trigger remediation if the node is NotReady for more than 10 seconds
      type: Ready
    - status: 'False'
      timeout: 10s
      type: Ready
status:
```

Note: For example, **matchLabels** for a m13 cluster is as follows and can be found in Compute > MachinesSet details.

- **machine.openshift.io/cluster-api-cluster=rackm13-rbzr6**
- **machine.openshift.io/cluster-api-machine-role=worker**
- **machine.openshift.io/cluster-api-machine-type=worker**
- **machine.openshift.io/cluster-api-machineset=rackm13-rbzr6-worker-0**
- When you configure a **MachineHealthCheck**, update the values under the **matchLabels** based on the following guidance:

  machine.openshift.io/cluster-api-cluster
  : Use cluster machineset name. To find the name, run the following command:

    ```
    oc get machineset -A
    ```

  machine.openshift.io/cluster-api-machine-role
  : Specify the machine role. For example, `master` or `worker`.

  machine.openshift.io/cluster-api-machine-type
  : Indicate the machine role. Use `worker`

  machine.openshift.io/cluster-api-machineset
  : Combine cluster name, a custom label, and the zone. The syntax is **<cluster_name>-<label>-<zone>**.

- The **maxUnhealthy** field controls when remediation is allowed, based on how many nodes are unhealthy in the cluster.**MaxUnhealthy** field in **MachineHealthCheck** specifies either an absolute number or a percentage of the total Machines checked by **MachineHealthCheck**.
  When **maxUnhealthy** is set to a fixed number, for example, `2`:
  - Remediation occurs only whenever the number of unhealthy nodes are within the allowed limit. It must not be more than two. For example, If the cluster has five nodes and **maxUnhealthy** is set to `2`, then whenever three nodes become unhealthy, remediation is not triggered.
  - If there are zero, one, or two unhealthy nodes, then remediation is allowed.
  - If there are three or more unhealthy nodes, then remediation is skipped.

  When **maxUnhealthy** is set as a percentage, for example, 40%:
  - The allowed number of unhealthy nodes is calculated as a percentage of total nodes. For instance, if a cluster consists of five nodes and **maxUnhealthy** is set to 40%, remediation is not be triggered when three nodes become unhealthy, as this matches the defined limit of 40% (3 nodes).
  - If there are unhealthy nodes ≤ 40% of the total nodes, then remediation is allowed.
  - If there are unhealthy nodes >40% of the total nodes, then remediation is skipped.

  The following table serves as a reference for the same example information:

maxUnhealthy Setting	**Cluster Size**	**Allowed Unhealthy Nodes**	**Remediation Triggered When**
2 (fixed number)	5 nodes	Up to 2 nodes	≤ 2 nodes unhealthy
40% (percentage)	5 nodes	Up to 2 nodes (40%)	≤ 2 nodes unhealthy

  Note: Remediation is triggered only when the count of unhealthy nodes remains within the threshold set by **maxUnhealthy**, whether specified as a fixed value or a percentage of the total nodes.
  This configuration ensures that the health check is applied only to the machines that match these specific labels.

Backup and restore of virtual machines

IBM Fusion protects virtual machines from data loss and corruption by backing up the virtual machines regularly. The virtual machines can then be restored from a previous backup.

Backup virtual machines

Backup policies define the frequency of backups, their storage locations, and the duration for which they must be retained. A best practice is to store backups in an object storage outside of the OpenShift® cluster, as this allows virtual machines to be recovered even when the original cluster is lost or compromised.

Virtual machine backups are incremental, meaning that only changed data is part of each backup. This approach reduces the amount of network traffic that is needed to transfer backups to object storage, and reduces the object storage requirements.

The IBM Fusion offers the following virtual machine backups through the Backup & Restore service.

In-place backup of virtual machine and Red Hat® OpenShift resources
: Virtual machines can be restored in place, or in a different Red Hat OpenShift cluster.

Option to copy backup to object storage
: Backups can be stored in object storage that is on a site or in the public cloud.

Integrated backup policy management
: Virtual machines are protected by using backup policies, which define the frequency of backups, their storage locations, and the duration for which they must be retained.

Incremental backup of virtual machine
: Only changes are included in each backup, reducing the amount of network traffic and object storage utilization.

To protect a group of virtual machines that represents an application, assign a backup policy to the namespace that the virtual machines belong to. It automatically backs up all the virtual machines in the namespace that uses the backup policy.
Note: A Red Hat OpenShift best practice is to create a namespace for a collection of virtual machines that represent an individual application.
To assign backup policies to an application that contains the virtual machines, go to the Applications page in the IBM Fusion user interface. Select the namespace that the virtual machines belong to and then assign a backup policy. Alternatively, you can manage backup policies declaratively by using the Custom Resources. It enables you to manage a large fleet of applications by using a GitOps approach, where a **`PolicyAssignment`** CR is deployed to the cluster alongside the application.

Restore virtual machines

To recover an application that is made up of virtual machines, select a backup to restore. Use the timestamp of each backup to select the appropriate point in time for restoration. The application can be restored in place, or you can restore the application to a different Red Hat OpenShift cluster.

IBM Fusion provides the following recovery options through its Backup & Restore service:

- You can recover a virtual machine to the same or alternative cluster.
- You can recover all virtual machines in a namespace to an alternative namespace and then delete the virtual machines that are no longer needed.
- You can recover all virtual machines in a namespace to an alternative namespace and then copy files from the target virtual machine by using the command line.
- You can manage the assignment of a backup policy to a high volume of virtual machines by directly working with the Fusion data protection custom resources, specifically the resources that are used to create backup policies and backup policy assignments.
- You can restore subsets of virtual machines that belong to an application by using the following two methods:
  - You can restore the entire application into an alternative namespace and then delete the virtual machines that are no longer needed.
  - You can restore the entire application into a namespace to an alternative namespace and then copy files from the target virtual machine by using the command line.

To validate, restart the virtual machine after recovery.

Achieving data consistency for virtual machine workloads

Achieving data consistency for virtual machines by using file system consistency, crash consistency, or application consistency approaches. For more information, see Achieving data consistency for virtual machine workloads.

Creating a virtual machine by using Clone PVC

A clone is an exact duplicate of an existing storage volume that can be used like any standard volume. You create a clone to capture a point-in-time snapshot of the data.

Procedure

1. Go to Virtualization > Virtual Machines.
2. Click Create virtual machine and select the option from template.
3. Select the virtual machine template used earlier to deploy on primary cluster and click disk source.
   If the option does not exist, then click Customise virtual machine.
4. Click Clone PVC and select the PVC project (virtual machine namespace).
5. Select the PVC name.
6. Click Create virtual machine.

Disaster recovery for virtual machines

Disaster Recovery of virtual machines involves creating regular backups, maintaining up-to-date replicas, and implementing robust failover mechanisms. Implementing failover mechanisms ensures high availability and minimizes potential data loss, reducing downtime, and ensuring long-term resilience against disasters.

About this task

The topic provides the procedure on how to enable protection for virtual machines through disaster recovery by using IBM Fusion.

Procedure

1. Set up Metro-DR configuration between two clusters.
2. Enable disaster recovery protection for the virtual machine.
   For the procedure to enable or enroll virtual machines for disaster protection, see Enabling and disabling applications for. After the application is enrolled for disaster recovery, go the virtual machine Overview tab page and check whether the Metro status is synchronized in the Disaster recovery section. Go to the Replicated applications page to see the virtual machine with details of the primary and partner cluster.
   Note: Before enabling disaster recovery protection for a virtual machine, remove any installation ISO volumes. These volumes are typically only required during the initial operating system installation and are not necessary for workload recovery. If an ISO volume must remain attached, reference the ISO persistent volume claim (PVC) directly using `persistentVolumeClaim` instead of creating an upload-based data volume.
3. From the user interface of the partner cluster, failover virtual machines.
   Important: As a prerequisite for virtual machine failover, shut down the virtual machine on the primary cluster.
   a. In the Red Hat® OpenShift® console of the partner cluster, go to Virtualization > VirtualMachines.
   b. Search for the virtual machine and from the ellipsis overflow menu, select Stop.
   Before you delete a virtual machine, make sure to copy the YAML file so you can redeploy the virtual machine using that at a failover site. If you stopped the virtual machine before copying the YAML, restart it after the deployment.

   For the procedure to failover, see Failover applications from site 1 to site 2.
4. After the failover is complete, remove the original virtual machine from the primary cluster.
   a. Go to the Red Hat OpenShift console of the original virtual machine on the primary cluster.
   b. Go to Virtualization > Virtual Machines.
   c. Search for the and from the ellipsis overflow menu, select Delete.
   d. In the Delete VirtualMachine window, click Delete.
   e. Select the resources of the virtual machine and click Delete.
5. Deploy the virtual machine on the current primary cluster.
   a. In the virtualMachines page of the Open shift console, click create virtual machine and select the option with YAML.
   b. Copy paste the YAML file copied from the original cluster.
      The status of the newly deployed virtual machine is in Running state.
   c. To validate, do the following steps:
      i. Go to Storage PersistenVolumeClaims and check whether the virtual machine is operational. Also, check whether the used capacity of the persistentvolumeclaim is correct.
      ii. Alternatively, check the Used (GiB) capacity column value of the virtual machine in the Applications page of user interface.
   Note: If a virtual machine is created using template catalogs, then you can also create Virtual Machine using Clone PVC option. For the procedure to clone, see Creating a virtual machine by using Clone PVC.
6. Initiate the planned failback.
   a. Stop the virtual machine in the current primary cluster.
   b. Go to the the user interface of the partner cluster and failover the virtual machine.
      For the procedure to The primary cluster reverts to the original primary where the virtual machine was set up originally.
   c. Remove the virtual machine from the current primary cluster.
7. Deploy the virtual machine on the partner cluster or the original primary cluster.
   a. In the virtualMachines page of the Open shift console, click create virtual machine and select the option with YAML.
   b. Copy paste the YAML file copied from the original cluster.
      The status of the newly deployed virtual machine is in Running state.
   c. To validate, do the following steps:
      i. Go to Storage PersistenVolumeClaims and check whether the virtual machine is operational. Also, check whether the used capacity of the persistentvolumeclaim is correct.
      ii. Alternatively, check the Used (GiB) capacity column value of the virtual machine in the Applications page of user interface.
   Note: If a virtual machine is created using template catalogs, then you can also create Virtual Machine using Clone PVC option. For the procedure to clone, see Creating a virtual machine by using Clone PVC.

Advanced workloads

IBM Fusion is built to provide enterprise ready deployments for IBM Cloud Paks, IBM watsonx, IBM Maximo®, and Db2 Warehouse.

- **IBM Cloud Paks support for IBM Fusion**
  Mapping of IBM Cloud Paks with IBM Fusion.
- **IBM Maximo® Applications support for IBM Fusion**
  Mapping of IBM Maximo® Applications with IBM Fusion.
- **IBM watsonx**
  The watsonx.data on-premise solution integrates with Red Hat OpenShift and uses IBM Storage Ceph for cloud-scale object storage and IBM Fusion for optimized application platform storage.
- **AI workloads on IBM Fusion**
  IBM Fusion combined with IBM watsonx offers a consistent high performance across a wide range of container AI and VM workloads.

IBM Cloud Paks support for IBM Fusion

Mapping of IBM Cloud Paks with IBM Fusion.

IBM Cloud Paks	Version	Backup and restore support	Storage support	Hosted Control Plane
IBM Software Hub (formerly Cloud Pak for Data) Note: IBM Software Hub supports backup and restore on both x86 and Power. IBM watsonx support in IBM Software Hub: • watsonx Assistant • watsonx.ai • watsonx Code Assistant • watsonx Code Assistant for Red Hat Ansible Lightspeed • watsonx Code Assistant for Z • watsonx Code Assistant for Z Code Explanation • watsonx.data • watsonx.governance • watsonx Orchestrate	5.3.x or later	• Online backup and recovery to the same or alternate cluster using IBM Fusion • Backup and restore to the same cluster using IBM Software Hub native tools	• Global Data Platform • Fusion Data Foundation • IBM Software Hub storage considerations • Red Hat® OpenShift® Data Foundation	Supported See IBM Software Hub cloud deployment environments
IBM Cloud Pak for Business Automation	24.0 or later	• Online backup and restore to same or alternate cluster using IBM Fusion • Backup and restore using IBM Cloud Pak for Business Automation native tools	• Global Data Platform • Fusion Data Foundation • IBM Cloud Pak for Business Automation storage considerations • Red Hat OpenShift Data Foundation	Not supported
IBM Cloud Pak for Integration	16.1.0 or later	• Backup and restore using IBM Cloud Pak for Integration native tools	• Global Data Platform • Fusion Data Foundation • IBM Cloud Pak for Integration storage considerations • Red Hat OpenShift Data Foundation	Supported See IBM Cloud Pak for Integration documentation
IBM Cloud Pak for AIOps	4.10.x or later	• Online backup and recovery to the same or alternate cluster using IBM Fusion • Backup and restore using IBM Cloud Pak for AIOps native tools	• Global Data Platform • Fusion Data Foundation • Storage considerations for IBM Cloud Pak for AIOps AI Manager • Red Hat OpenShift Data Foundation	Supported See IBM Cloud Pak for AIOps documentation
IBM Cloud Pak for Security	1.10 or later	Backup and restore using IBM Cloud Pak for Security native tools	• Global Data Platform • Fusion Data Foundation • IBM Cloud Pak for Security storage considerations • Red Hat OpenShift Data Foundation	Not supported
IBM Cloud Pak for Network Automation	2.6.x or later	Backup and restore using IBM Cloud Pak for Network Automation native tools	• Global Data Platform • Fusion Data Foundation • IBM Cloud Pak for Network Automation storage considerations • Red Hat OpenShift Data Foundation	Not supported

For the storage class to use with IBM Cloud Paks, IBM Storage Scale version (All), see Storage class to use with Cloud Paks.

To use IBM Cloud Paks in IBM Fusion, the storage class must include `shared: true`.

- **Validation tools for IBM Cloud Paks**
  Run the storage readiness and performance suites on your IBM Fusion environments. The suites run on OpenShift Container Platform to evaluate the setup and performance of your storage setup.

Validation tools for IBM Cloud Paks

Run the storage readiness and performance suites on your IBM Fusion environments. The suites run on OpenShift® Container Platform to evaluate the setup and performance of your storage setup.

Storage validation tool

Run the IBM Software Hub storage validation tool on your Red Hat® OpenShift cluster to verify your storage setup for use with IBM Software Hub. Set up your environment and before you install your IBM Cloud Paks, run the validation tool. For more information about the tool, see storage validation tool.

Performance validation tool

Run the storage performance tool to ensure that your storage performs properly and you meet the recommended performance guidelines. Collect Storage performance metrics on your Red Hat OpenShift cluster. For more information about the tool, see k8s-storage-perf GitHub repository.

As a prerequisite to run storage performance suite, run the following command to install selinux:

```
pip install selinux
```

The tool validates whether your performance (for example, network, environment) runs at a speed that enables IBM Cloud Paks to run successfully.
An example of the output:

```
Summary
Cluster Name                PVC               Storage Type  Environment     Test Name   Thread Count  write MiB/s          Require
storage-performance-cluster pvc-sysbench-rwo  Scale           ibmcloud      rndwr_4k_8        8             46.4         Recommende
storage-performance-cluster pvc-sysbench-rwx  Scale           ibmcloud      seqwr_1g_2        2             3766.0       Recommende
storage-performance-cluster pvc-sysbench-rwo  Scale           ibmcloud      seqwr_1g_2        2             3901.2       Recommende
storage-performance-cluster pvc-sysbench-rwx  Scale           ibmcloud      rndwr_4k_8        8             46.1         Recommende
```

Here, the PVCs are running against the tests. The thread count specifies the number of simultaneous requests. The write Mb/s indicates the output.

Example of the partial output (from the "all" full metrics run) of a minimum IBM Fusion HCI configuration, that is, for a smallest environment. Depending on your configuration, environment, and network speed, the output would vary.

Summary

Cluster Name	PVC	Storage Type	Environment	Test Name	Thread Coun	write MiB/s	Requirement
storage-performance-cluster	pvc-sysbench-rwo	odf	ibmcloud	rndwr_4k_8	8	30.8	Recommended to meet the requirement of 11 MiB/s or higher
storage-performance-cluster	pvc-sysbench-rwo	odf	ibmcloud	seqwr_1g_2	2	4348.9	Recommended to meet the requirement of 128 MiB/s or higher
storage-performance-cluster	pvc-sysbench-rwx	odf	ibmcloud	rndwr_4k_8	8	32.1	Recommended to meet the requirement of 11 MiB/s or higher
storage-performance-cluster	pvc-sysbench-rwx	odf	ibmcloud	seqwr_1g_2	2	4463	Recommended to meet the requirement of 128 MiB/s or higher

Detailed Measurements

Cluster Name	PVC	Storage Type	Environment	Test Name	Thread Coun	write MiB/s	Writes/s	read MiB/s	Reads/s	Total Time	Latency Min	Latency Avg	Latency Max	Latency 95th
storage-performance-cluster	pvc-sysbench-rwo	odf	ibmcloud	rndwr_4k_8	8	30.8	7876.3	0	0	10	0.5	1	110.5	1.5
storage-performance-cluster	pvc-sysbench-rwo	odf	ibmcloud	seqwr_1g_2	2	4348.9	4.2	0	0	10.3	308.2	472.2	735.4	592.5
storage-performance-cluster	pvc-sysbench-rwx	odf	ibmcloud	rndwr_4k_8	8	32.1	8213	0	0	10	0.5	0.9	40.1	1.5
storage-performance-cluster	pvc-sysbench-rwx	odf	ibmcloud	seqwr_1g_2	2	4463	4.3	0	0	10.2	283.4	458.4	698.1	556.5

Here, the cluster name is `storage-performance-cluster` of storage type `odf`. The *Summary* section of the table also includes the requirement recommendations. In this example, for the `rndwr_4k_8`, the write MiB/s is 30.8 and the recommended is 11 MiB/s or higher. In all the rows, the output is either 2X or more than 2X of the recommendation. It helps you validate whether your environment meets the recommended performance requirements for IBM Cloud Paks.

You can use this feature regardless of IBM Cloud Paks.

IBM Software Hub sizing in the Sales Configurator

The starting point is a minimum IBM Software Hub configuration. If you select services from the list for inclusion, it shows the minimum required configuration. For example, it displays the number of compute nodes, resource pools, and memory to deploy. It also lists the dependencies. For example, if you want to use Watson OpenScale, then you need Watson Machine Learning, Common Core Services, and Analytics Engine Powered by Apache Spark.
Note: Ensure that the IBM Fusion/storage sizing are also considered in your IBM Software Hub sizing.
To view the sales configurator, go to https://app.ibmsalesconfigurator.com/#/zen/configure.

IBM Maximo® Applications support for IBM Fusion

Mapping of IBM Maximo® Applications with IBM Fusion.

IBM Maximo® Applications	Version	Backup & Restore support	Storage support	Hosted Control Plane
Maximo Core	9.1.x	Online backup and restore for Maximo Core using IBM Fusion	• Global Data Platform • Fusion Data Foundation • Red Hat® OpenShift® Data Foundation	• Supported on Bare Metal Hosted Control Plane • Supported on Virtualized Hosted Control Plane For more information about Hosted Control Plane with IBM Fusion, see IBM Maximo® on Hosted Control Plane.
Maximo Manage	9.1.x	Online backup and restore for Maximo Manage using IBM Fusion	• Global Data Platform • Fusion Data Foundation • Red Hat OpenShift Data Foundation	• Supported on Bare Metal Hosted Control Plane • Supported on Virtualized Hosted Control Plane For more information about Hosted Control Plane with IBM Fusion, see IBM Maximo® on Hosted Control Plane.

IBM Maximo® Applications	Version	Backup & Restore support	Storage support	Hosted Control Plane
Maximo Optimizer	9.1.x	Online backup and restore for Maximo Optimizer using IBM Fusion	• Global Data Platform • Fusion Data Foundation • Red Hat OpenShift Data Foundation	• Supported on Bare Metal Hosted Control Plane • Supported on Virtualized Hosted Control Plane For more information about Hosted Control Plane with IBM Fusion, see IBM Maximo® on Hosted Control Plane.
Maximo Health	Deployed as part of Manage	Online backup and restore for Maximo Health using IBM Fusion	• Global Data Platform • Fusion Data Foundation • Red Hat OpenShift Data Foundation	• Supported on Bare Metal Hosted Control Plane • Supported on Virtualized Hosted Control Plane For more information about Hosted Control Plane with IBM Fusion, see IBM Maximo® on Hosted Control Plane.
Maximo IOT	9.1.x	Online backup and restore for Maximo IOT using IBM Fusion	• Global Data Platform • Fusion Data Foundation • Red Hat OpenShift Data Foundation	• Supported on Bare Metal Hosted Control Plane • Supported on Virtualized Hosted Control Plane For more information about Hosted Control Plane with IBM Fusion, see IBM Maximo® on Hosted Control Plane.
Maximo Monitor	9.1.x	Online backup and restore for Maximo Monitor using IBM Fusion	• Global Data Platform • Fusion Data Foundation • Red Hat OpenShift Data Foundation	• Supported on Bare Metal Hosted Control Plane • Supported on Virtualized Hosted Control Plane For more information about Hosted Control Plane with IBM Fusion, see IBM Maximo® on Hosted Control Plane.
Maximo Predict	9.1.x	Online backup and restore for Maximo Predict using IBM Fusion	• Global Data Platform • Fusion Data Foundation • Red Hat OpenShift Data Foundation	• Supported on Bare Metal Hosted Control Plane • Supported on Virtualized Hosted Control Plane For more information about Hosted Control Plane with IBM Fusion, see IBM Maximo® on Hosted Control Plane.
Maximo Collaborate (formerly Assist)	9.1.x	Online backup and restore for Maximo Collaborate (formerly Assist) using IBM Fusion	• Global Data Platform • Fusion Data Foundation • Red Hat OpenShift Data Foundation	• Supported on Bare Metal Hosted Control Plane • Supported on Virtualized Hosted Control Plane For more information about Hosted Control Plane with IBM Fusion, see IBM Maximo® on Hosted Control Plane.
Maximo Visual Inspection Note: To check supported GPU models, see Maximo Visual Inspection documentation.	9.1.x	Online backup and restore for Visual Inspection using IBM Fusion	• Global Data Platform • Fusion Data Foundation • Red Hat OpenShift Data Foundation	• Supported on Bare Metal Hosted Control Plane • Supported on Virtualized Hosted Control Plane For more information about Hosted Control Plane with IBM Fusion, see IBM Maximo® on Hosted Control Plane.

For the general Maximo BR procedure, see https://github.com/IBM/storage-fusion/tree/master/backup-restore/recipes/Maximo/2.9.x.

- **IBM Maximo® Application Suite**
  IBM Maximo® Application Suite is a collection of AI-driven asset management applications that gains enhanced backup, restore, and storage capabilities when deployed with IBM Fusion.
- **IBM Maximo® on Hosted Control Plane**
  Use the IBM Maximo® IBM Operator catalog and monthly updates for the latest information about the Fusion operator.

IBM Maximo® Application Suite

IBM Maximo® Application Suite is a collection of AI-driven asset management applications that gains enhanced backup, restore, and storage capabilities when deployed with IBM Fusion.

IBM Maximo® Application Suite provides powerful AI-driven asset and analytics solutions for your enterprise needs.

IBM Maximo® Application Suite is a powerful collection of applications, industry solutions, add-ons, and tools.

When you deploy it from IBM Fusion, it enhances backup, restore, and storage capabilities across IBM Maximo® Application Suite.

If you have the IBM Maximo® software configured in your environment or have plans to configure it, then use only the **ibm-operator-catalog** of the IBM Maximo® software. For more information to install this catalog, see IBM Catalogs and Installing IBM Maximo® Application Suite.

Install

The IBM Maximo® CLI is the preferred way to install IBM Maximo® as it provides an automated deployment for this type of workload. For more information, see IBM Maximo® KC and IBM Maximo® Application Suite CLI.
Tip: To customize IBM Maximo® Application Suite instances with extra features that might be unavailable in IBM Maximo® Application Suite CLI, you can use ansible-devops for advanced configuration. For more information, seeMAS Devops Ansible Collection.
For installation sizing calculator guidance, see Sizing guidance.

Hosted Control Plane support

To install IBM Maximo® Application Suite on a Hosted Control Plane, see IBM Maximo® on Hosted Control Plane.

Limitations

If you plan to use any IBM Maximo® application dependent on Kafka such as IOT and Monitor, do not use the Strimzi open source operator in conjunction with IBM Fusion Backup & Restore. Instead, use the Streams for Apache Kafka by Red Hat® (formerly AMQ-Streams). Ensure that you match the version available in the IBM Fusion Backup & Restore namespace with the Streams operator that you are planning to use for IBM Maximo® Application Suite. For more information about the limitations, see IBM Maximo® limitations.

Backup & Restore is not supported for IBM Maximo® IoT, IBM Maximo® Monitor, and IBM Maximo® Predict, but they can be installed on any cluster that has OpenShift® Virtualization installed.

IBM Maximo® on Hosted Control Plane

Use the IBM Maximo® IBM Operator catalog and monthly updates for the latest information about the Fusion operator.

For the list of IBM Maximo® deployments, see IBM Maximo® Applications support for IBM Fusion.

If you plan to use IBM Maximo® in Hosted Control Plane with IBM Fusion you must deploy Fusion operator and Fusion Services by logging into the Hosted Control Plane cluster and use the following procedure:

- Install IBM Fusion on Bare Metal. See Installing IBM Fusion on On-premises Bare Metal.
- Install Fusion Data Foundation. See Installing Fusion Data Foundation on the hosted cluster.
- Install Backup & Restore Spoke. See Backup & Restore .

Install

The CLI is the preferred way to install as it provides an automated deployment for this type of workload. For more information, see Maximo KC and MAS CLI.
Tip: To customize IBM Maximo® Application Suite instances with extra features that might be unavailable in IBM Maximo® Application Suite CLI, you can use ansible-devops for advanced configuration. For more information, see Maximo Application Suite Devops Ansible Collection.
For installation sizing calculator guidance, see Sizing guidance.

IBM watsonx

The watsonx.data on-premise solution integrates with Red Hat® OpenShift® and uses IBM Storage Ceph for cloud-scale object storage and IBM Fusion for optimized application platform storage.

watsonx.data is an open, hybrid, and governed data lakehouse optimized for all data and AI workloads. It offers a single point of entry where you can store the data or attach data sources for managing and analyzing your enterprise data (structured, semi-structured, and unstructured), enabling access to all data across cloud and on-premise environments.

The watsonx.data is provided as part of IBM Software Hub (formally IBM Cloud Pak for Data), which is a modular set of integrated software components for data analysis, organization, and management. Red Hat OpenShift Container Platform provides a container run time for watsonx.data. Red Hat OpenShift Container Platform is hosted on bare metal x86 servers. IBM Storage Ceph is a software defined storage platform based upon an open source development model and deploys on industry-standard x86 hardware. It provides non-disruptive, horizontal scaling of object, block, and file storage with access to large capacities of data (petabytes to exabytes), ideal for modern AI frameworks that require data lake capabilities.

As part of this solution, IBM Fusion acts as a storage provider for the IBM Software Hub platform that hosts watsonx.data. IBM Fusion exposes storage classes that are used to provide RWO and RWX persistent volumes that IBM Software Hub needs to run. IBM Fusion obtains storage by remotely mounting the ESS 3500 file system using the Global Data Platform service. When Container Storage Interface (CSI) requests are made, the Global Data Platform provisions storage from the ESS 3500 file system and provides persistent volumes to the container workload.

IBM Fusion also provides watsonx.data with S3 access to the storage acceleration cache on the ESS 3500. Global Data Platform creates a statically provisioned volume that maps to the ESS 3500 fileset that acts as a cache for an accelerated object bucket. IBM Fusion then uses a technology called the Multi-cloud Gateway to expose the cache through S3 protocols. The watsonx.data attaches to the S3 bucket provided by the Multi-cloud Gateway, enabling query engines to take advantage of the storage acceleration cache.

AI workloads on IBM Fusion

IBM Fusion combined with IBM watsonx offers a consistent high performance across a wide range of container AI and VM workloads.

Node labeling and GPU virtualization techniques can effectively enhance performance and ensure efficient resource management for both complex AI container workloads and virtual machine (VM) workloads.

Node labeling for GPU reservation

IBM Fusion uses labels to reserve GPU nodes for AI workloads.

For example, the IBM Fusion Backup & Restore uses the `isf.ibm.com/nodeType: gpu` label to control the placement of its pods. To prevent Backup & Restore workloads on GPU nodes, run the following command from the OpenShift® cluster to label the GPU nodes.

```
oc label node/<node name> gpu.isf.ibm.com=""
```

The IBM Fusion service pods avoid nodes with such a label. You can use this approach for other workloads.
For example, the IBM Fusion Backup & Restore uses the `gpu.isf.ibm.comisf.ibm.com/nodeType: gpu` label to manage the placement of its pods effectively. To restrict IBM Fusion Backup & Restore workloads from running on GPU nodes, run the following command to apply this label to these nodes in the OpenShift cluster. After the nodes are labeled, the Backup & Restore pods automatically avoid GPU nodes with this label. This approach can also be adapted for other workloads to ensure resource optimization and placement control.

Administering

You can manage and administer hardware, monitor and collect logs, understand events, raise call tickets, work with APIs, manage users, and understand the UI.

- **Observability**
  Observability in IBM Fusion enables centralized, multi-cluster monitoring that uses open source technologies like Prometheus, Thanos, and Grafana. This solution collects metrics from managed clusters, provides global querying and long-term storage, and delivers unified dashboards for visualization and alerting.
- **Knowing your IBM Fusion user interface**
  Learn the various key elements of the IBM Fusion user interface for ease of use and navigation.
- **Upgrading IBM Fusion**
  Upgrade to the latest version so you can have access to the new functionalities and improvements that enhance your overall experience with IBM Fusion.

Observability

Observability in IBM Fusion enables centralized, multi-cluster monitoring that uses open source technologies like Prometheus, Thanos, and Grafana. This solution collects metrics from managed clusters, provides global querying and long-term storage, and delivers unified dashboards for visualization and alerting.

- **AI-enabled observability with Red Hat OpenShift Lightspeed**
  Simplify observability across clusters through smart agents.
- **IBM Fusion metrics**
  This section describes the metrics that are used in IBM Fusion.
- **Enabling observability for IBM Fusion clusters**
  Use these instructions to enable multi-cluster observability for IBM Fusion clusters.
- **IBM Fusion Call Home Telemetry**
  IBM Fusion Call Home Telemetry enhances supportability by providing IBM with valuable insights into system configuration and performance. Telemetry data is collected from various components and securely transmitted to IBM, helping support teams proactively identify issues and optimize system behavior based on real-world usage.

AI-enabled observability with Red Hat OpenShift Lightspeed

Simplify observability across clusters through smart agents.

Red Hat OpenShift® Lightspeed provides an AI-based chat window that understands OpenShift and the current cluster state. When integrated through the Model Context Protocol (MCP), the chat window retrieves live data from the cluster when configured. As a result, responses reflect the current state rather than static documentation. You interact exclusively through the chat window: you submit a query, Lightspeed uses MCP to retrieve the required data, and then explains and summarizes the results. This approach provides a single location for monitoring health, troubleshooting issues, and reviewing summarized information without switching between the OpenShift console and other tools.

One place to ask questions
: Use the Lightspeed chat window, either in the OpenShift console or from an IDE through MCP, to ask about cluster health, workloads, and issues. The chat window uses MCP to access live cluster state and, when available, metrics and alerts, providing a consolidated conversational view.

On-demand debugging and analysis
: Ask questions such as *What's wrong with the cluster?*, *Why is this failing?*, or *Show me what's running in this namespace*. The LLM retrieves current data through MCP and combines it with the knowledge of OpenShift to provide focused, context-aware responses without running anything in the background.

Summarized insights
: Request high-level overviews in plain language, such as *Summarize what's unhealthy* or *What's going on in this project?* The chat window used MCP to retrieve the relevant resources and return concise summaries, enabling quick triage and deeper investigation only when required.

Integration with existing tools
: MCP complements existing monitoring tools rather than replacing them by adding a chat window on top. Continue to use Grafana for dashboards and Prometheus for metrics, while using Lightspeed for quick queries, summaries, and guided troubleshooting based on the same data.

IBM Fusion metrics

This section describes the metrics that are used in IBM Fusion.

Platform

Metric	Description
`isf_cluster_details`	It displays the details about the cluster (version, platform, and environment).
`isf_service`	It displays the health status of the installed services ('1' for healthy, '0' for unhealthy).
`isf_environment_risk`	It displays the count of active environment risks on the cluster.
`isf_service_upgrade_risk_count`	It displays the count of active upgrade pre-check risks per service on the cluster.

Fusion events

Metric	Description
`fusion_events_total`	It displays the count of total unique fusion events created.
`fusion_duplicate_events_total`	It displays the count of total fusion events that are identified as duplicate.
`fusion_hardware_events_total`	It displays the count of total fusion events that are associated with a hardware asset.

Storage

Metric	Description
`isf_local_storage_total_gib`	It displays the local storage type and its usable capacity metric in GiB.
`isf_local_storage_used_percentage`	It displays the percentage of local storage used.

Enabling observability for IBM Fusion clusters

Use these instructions to enable multi-cluster observability for IBM Fusion clusters.

Before you begin

Ensure that you met the following prerequisites:

- Network connectivity between clusters and the hub or query endpoint.
- S3-compatible object storage for long-term metrics retention.

About this task

Achieve a comprehensive multi-cluster observability solution for IBM Fusion by using Red Hat® Advanced Cluster Management or an existing Thanos and Grafana infrastructure. Multi-cluster observability uses open source technologies to create a centralized monitoring solution:

- `Thanos` - Provides global querying across multiple Prometheus instances and long-term metrics storage in S3-compatible object stores.
- `Grafana` - Delivers visualization dashboards for metrics analysis and alerting.
- `Prometheus` - Collects metrics from each managed cluster and sends data to the hub cluster.

It delivers **centralized fleet-wide monitoring** through unified dashboards, enabling proactive issue identification and management across the following IBM Fusion components:

- Fleet overview
- Unhealthy data services
- Active environmental risks
- Event metrics
- Upgrade risks for IBM Fusion services
- Storage cluster health

Procedure

Resources for custom IBM Fusion observability dashboards or configuration YAMLs are available through the GitHub repository.

What to do next

The centralized dashboard represents a fundamental shift from reactive to proactive infrastructure management, ensuring your IBM Fusion fleet maintains optimal performance while minimizing the risk of undetected issues escalating into service outages.

IBM Fusion Call Home Telemetry

IBM Fusion Call Home Telemetry enhances supportability by providing IBM with valuable insights into system configuration and performance. Telemetry data is collected from various components and securely transmitted to IBM, helping support teams proactively identify issues and optimize system behavior based on real-world usage.

Telemetry automatically collects and transmits the data from IBM Fusion components, such as software applications or hardware devices. Also, it is used to gather information about system configuration, usage patterns, and performance metrics. For steps to enable call home telemetry, see Enabling Call Home.
Note: The system uploads Call Home telemetry data every 24 to 26 hours.
The following details are collected and uploaded as part of call home telemetry support:

IBM Fusion cluster details
: Collects statistical insights into IBM Fusion resources, such as the number of pods, deployments, and other key components, along with version and health status details for both OpenShift® and IBM Fusion. Also, it collects node storage consumption details for OpenShift.

IBM Fusion license data
: Collects IBM Fusion licensing data. For more information on licensing metrics, see Understanding the IBM Fusion license report.

IBM Fusion events data
: Collects the summary of IBM Fusion events by severity and event ID.

IBM Fusion Services
: Collects the details such as name, version, health, and upgrade details of different fusion services installed in the cluster. If Backup & Restore service is installed, then it shows the success % of backups and restores.

Hosted clusters
: Collects basic count and health status of hosted clusters that are deployed.

Virtual machines
: Collects basic count and health status of virtual machines that are created on cluster.

Knowing your IBM Fusion user interface

Learn the various key elements of the IBM Fusion user interface for ease of use and navigation.

To open the user interface from any of the supported browsers:

1. Log in to OpenShift® Container Platform using kubeadmin.
2. Click the Applications icon in the title bar and click IBM Fusion.
3. In the IBM Fusion user interface, click Let's get started in the Welcome to IBM Fusion page.

The Quick start page is the landing page of IBM Fusion user interface.
From this Quick start page, you can provide storage for container workloads, backup and restore the applications and control plane data, upgrade components, investigate events and open support tickets.

The following options are available in the landing page of IBM Fusion user interface:

- You can do the following tasks from the navigation menu:
  - Click Events to view IBM Fusion events. They are Kubernetes native events that include filter-specific labels. For more information about the events page, see Analyzing events in IBM Fusion
  - Click Applications to backup and restore application and its resources. For more information about how to backup and restore applications, see Backup & Restore of your applications and virtual machines.
  - Click Backup policies to define parameters that are applied to backup jobs. For steps to create new policies or work with existing policies, see Backup policies.
  - Click Settings to do the following:
    - IBM Fusion
      You can set up encrypted data at rest by connecting to a Security Key Lifecycle Manager. For more information about how to set up encryption, see Configuring encryption for Global Data Platform storage.
- From the applications icon in the title bar, you can quickly navigate to the following consoles by using their corresponding outbound arrows:
  - Click OpenShift console to go to the OpenShift web management console. From the title bar of OpenShift console, click Red Hat applications > IBM Fusion to navigate to IBM Fusion.
- The following details and options are available in the title bar:
  - Displays the cluster ID of IBM Fusion.
  - Click the help icon on the title bar of the page to do the following tasks:
    - Click Open support case to create a support ticket with IBM.
    - Click Collect support logs to download support logs.
    - Click IBM Documentation to view IBM Documentation for IBM Fusion.
    - Click About to know the version of IBM Fusion.
  - The bell icon is visible on the IBM Fusion user interface only when any event or alert occurs and is available for the user. You can also click View all events to go the Events page.
  - Click the user icon to do the following tasks:
    - View the currently logged in user. Click user to view and manage the profile details of the logged in user.
    - Click Logout to log out of the IBM Fusion.
      Note: To close all open sessions in IBM Fusion, log out of both IBM Fusion UI console and OpenShift Container Platform UI console.

- **Working with the dashboard**
  Dashboard provides an overview of the infrastructure, managed clusters, hardware status, services health, storage, backup jobs, and environmental risks of the IBM Fusion.

- **Browser requirements**
  List of supported web browsers for IBM Fusion user interface.
- **Icons used in the user interface**
  A reference to the icons used in the user interface and their naming convention.
- **IBM Fusion user interface issues**
  A list of all known issues in the IBM Fusion user interface.

Working with the dashboard

Dashboard provides an overview of the infrastructure, managed clusters, hardware status, services health, storage, backup jobs, and environmental risks of the IBM Fusion.

Account
: The Account menu includes the following information:

  - IBM Fusion license and audit metrics report. Go to Account > License usage and click the download icon to get the detailed license and audit metrics report. To understand the metrics, see Understanding the IBM Fusion license report.

Details
: The Details section includes the following hardware information:

  - Cluster ID details.
  - Infrastructure details.
  - OpenShift® version details. Click Upgrade pre-check next to OpenShift version to open a slide out pane Upgrade pre-check to find the details of issues during the OpenShift upgrade. To debug further, click the outbound arrows to find exact problem details and possible prevention steps.
    Important: The upgrade pre-check identifies issues to be resolved before an upgrade is attempted, not the results of an upgrade operation.
  - Total number of nodes added to the cluster.

Services
: The Services section includes the following information:

  - It provides a detailed list of services that are installed on the IBM Fusion and their health status.
  - Click View services to go to Services page to view the list of available and installed services on the IBM Fusion along with their health status.

Backup jobs over time
: The Backup jobs over time section includes the total number of backups available.
  Important:

  - Ensure that the Backup & Restore service is installed to view backup jobs details. To get steps to install Backup & Restore service, click View Backup & Restore.
  - Ensure that a local storage service (either Data Foundation or Global Data Platform) is installed.
  - The local storage must be configured.

Local storage
: The Local storage section provides a circular pie chart that is used versus the available capacity of the storage. The storage capacity is represented in GiB and the Used capacity is represented graphically and in percentage. Click View storage to go to Local storage page to view the usable capacity, services health, and storage nodes details.
  Important:

  - Ensure that the Data Foundation service is installed.
  - The local storage must be configured.

- **Understanding the IBM Fusion license report**
  Use these instructions to get a detailed explanation of each metric and how to interpret them to ensure compliance with IBM Fusion licensing terms.

Understanding the IBM Fusion license report

Use these instructions to get a detailed explanation of each metric and how to interpret them to ensure compliance with IBM Fusion licensing terms.

IBM Fusion licenses require entitlements for the following key components:

- Cores managed
- Threads per core (SMT)
- Backup subscribed storage
- Services deployed
- Storage capacity provisioned or accessed

Cores managed

- Cores on all applicable nodes in the cluster must be entitled by using IBM Fusion licenses with a VPC metric or a managed server metric. For IBM Fusion HCI, for calculating the metric, the `PROCESSOR_CORE` metric considers all nodes by default, whereas in IBM Fusion, only worker nodes are considered.
- The reported value of `PROCESSOR_CORE` is obtained from a native OpenShift® API for each node and aggregated by the IBM Fusion license service.
- The quantity of cores that are included in IBM Fusion licenses with a compute metric must meet or exceed the `PROCESSOR_CORE` reported value.
- Licensing applies across all supported architectures, including x86, IBM Power, and IBM z. The processor core metric uses the name `PROCESSOR_CORE` (total vCPU count) for x86 nodes, `PROCESSOR_CORE_PPC64LE` (total vCPU count) for IBM Power nodes, and `PROCESSOR_CORE_S390X` (total vCPU count) for IBM z nodes.

IBM clients who purchase certain IBM Software products, such as IBM Cloud Pak, Watson, or Maximo Application Suite, may be entitled to use IBM Fusion as a supporting program within their license agreement.

Key entitlement details:

- Restrictions may apply to services and storage capacity, but not to the number of cores, unless otherwise stated.
- Clients using IBM Fusion as a supporting program must replace their entitlement with paid licenses to remove the restrictions.

Deployment requirements:

- IBM Fusion licenses with a managed server metric can only be deployed on x86 bare metal nodes.
- IBM Fusion licenses with a VPC metric can be deployed on any node type, including physical, virtual, on-premises, or public cloud.

Threads per core (SMT)
: The value reflects the maximum value of threads per core among all the nodes in the environment. The metric is reported in the licensing report and named as `NODE_SMT_MAX`.

Backup subscribed storage
: This value helps understand the total storage capacity, which is covered with backup and assigned a backup policy. The metric is reported in the licensing report and named as `BACKUP_STORAGE_GB`, and the value is reported in GiB.
Note: If two clusters are connected and configured as a hub and spoke, for spoke applications that are backed up on the hub, the metric value does not reflect the total storage capacity.

Services deployed

- Clients can install services from IBM Fusion using operators that are included in the product.
- Each service flag should be validated against the Fusion licenses purchased or applied to the observed cluster to ensure compliance.
- Restrictions of use in other products, such as IBM Cloud Paks, are applicable if IBM Fusion is deployed using terms from those products.

When a client installs services from IBM Fusion, they must ensure that the services are licensed for use. Each installed service has a flag with a value of 1. To validate, check the flag against the purchased or applied IBM Fusion licenses for the cluster.
Key considerations:

- Validate each service flag against the applicable license terms.
- If using IBM Cloud Paks, follow the restrictions of use for those products.
- Non-compliance may occur if services are used without proper licensing.

For example, if the BACKUP_RESTORE_SERVICE metric is 1 and IBM Cloud Pak license terms are used, the deployment may not be compliant.

Storage capacity provisioned or accessed

- IBM Fusion licenses usually include an allowance for provisioning or accessing storage in the cluster such as Fusion Data Foundation, IBM Storage Scale, IBM Storage Ceph Pro.
  Other types of storage can be accessed using Fusion services such as object storage, or a third-party storage array with Fusion Access.
- Each type of permitted provisioning or access method is reported in a metric with the prefix STORAGE_.
  For example, `STORAGE_FDF` can be provisioned within the cluster (locally) or provisioned outside the cluster (externally).
- Clients should validate that the storage capacities and types that are deployed are covered by IBM Fusion licenses and any product-specific licenses.
  For example, client may already own IBM Storage Scale and then add to it using their Fusion entitlements when setting up their Fusion cluster. Combining licenses for the same type of IBM storage from different sources is usually permitted.
- If the cluster is configured with the GDP remote mount metric, the `STORAGE_GDP_REMOTE_MOUNT` metric reports the total usable capacity in GiB.

Some Fusion licenses include an entitlement for primary storage and allow core deployment. If this entitlement is not enough, you can buy additional primary storage expansion parts to increase the capacity.
Primary storage:
Primary storage refers to the usable capacity available to a cluster.

IBM Software Entitlements:
If you are using IBM Software entitlements with limited storage capacity, you can increase your storage entitlements by replacing the included licenses with paid Fusion licenses.

- Replace the licenses with paid Fusion licenses that meet the processor core requirements.
- Buy additional capacity entitlements to cover any remaining storage needs.

Browser requirements

List of supported web browsers for IBM Fusion user interface.

In general, IBM Fusion user interface works with most of the modern web browsers. The following table lists the browsers that are tested against IBM Fusion user interface.Keep your browser version up to date with at least the last two versions so that the user interface works without issues.

Browsers	Supported
Mozilla Firefox	Supported
Google Chrome	Supported
Safari	Supported
Microsoft ® Edge	Supported

Icons used in the user interface

A reference to the icons used in the user interface and their naming convention.

Table 1. Symbol references

Icon	Naming convention used
	Green tick mark to indicate healthy, completed,
	Warning
	Critical, Not connected, Failed transferring data
	Canceled
	Search filter
	Information
	Launch YAML
	Restore
	Ellipsis overflow menu
	Settings
	External link
	Outbound arrow
	Run command for switch

IBM Fusion user interface issues

A list of all known issues in the IBM Fusion user interface.

- In a 100% browser resolution view, vertical scroll may not work as intended in the remote file system edit page. To access the fields outside the scroll area, zoom out of the browser page.

Upgrading IBM Fusion

Upgrade to the latest version so you can have access to the new functionalities and improvements that enhance your overall experience with IBM Fusion.

To achieve the upgrade, you must follow these steps in sequence:

1. Upgrade IBM Fusion software to the most recent version.
2. Upgrade all services in the specified sequence. For more information about the service upgrades, see Upgrading IBM Fusion services.

Each step builds on the previous one, so ensure they are performed in the correct order for a successful upgrade. Before you begin each of the upgrades, thoroughly review the prerequisites, support matrix, limitations, and known issues.
The IBM Fusion software is upgraded from the OpenShift® console.

You can upgrade the services from the Services page or the Settings > Upgrades page of the IBM Fusion user interface.

When you upgrade from the Services page, you can initiate upgrade of individual services based on the availability of a new version. In the Upgrade page, you can initiate multiple non-disruptive and rolling upgrades of service in parallel. For both non disruptive and rolling upgrades, you can view the status of the prechecks and monitor the progress of the upgrade. In the Services page, click View upgrade details of a ongoing service upgrade and in the Upgrade page, you can click the service tile of an ongoing upgrade to monitor the pre-check results and the progress of the upgrade in a slide out pane.

- **Prerequisites for enterprise registry upgrade**
  If you installed the earlier version of IBM Fusion by using your enterprise registry, then mirror the 2.12.0 images to your enterprise registry.
- **Upgrading IBM Fusion**
  This topic outlines the steps to upgrade IBM Fusion management software, Fusion services, and Red Hat® OpenShift Container Platform.
- **Upgrading Hosted Control Plane Fusion and its services**
  When IBM Fusion and services on the Hosted Control Plane cluster are installed using labels, they are automatically upgraded to the latest supported version available on the IBM Fusion hub.
- **Upgrading OpenShift Container Platform**
  Upgrade OpenShift in IBM Fusion. For offline upgrades of OpenShift Container Platform, follow the Red Hat steps. However, follow the guidelines and versions that are specified in this section.

- **Upgrading IBM Fusion services**
  From the IBM Fusion user interface, upgrade the IBM Fusion services, namely Data Foundation, Global Data Platform, and IBM Data Cataloging. The Backup & Restore service auto upgrades based on availability and you can monitor the progress.

Prerequisites for enterprise registry upgrade

If you installed the earlier version of IBM Fusion by using your enterprise registry, then mirror the 2.12.0 images to your enterprise registry.

1. Update the global pull secret with the mirror registry credentials to which you want to mirror the current version images. If you want to mirror to the same enterprise registry that you used in the previous version, then skip this step.
2. Use the end-to-end mirroring procedure to download all the images of IBM Fusion HCI and its services at once or individually depending on your specific needs. For individual mirroring procedures, see the following topics:
   - Mirror IBM Fusion images. See End-to-end mirroring of IBM Fusion and its services images to the enterprise registry.
   - Mirror Backup & Restore images. See Mirroring Backup & Restore images.
   - Mirror IBM Storage Scale images. See Mirroring IBM Storage Scale images.
   - IBM Data Cataloging offline upgrade:
     - Complete steps 1 to 2 of Mirroring IBM Data Cataloging images procedure.
     - Update the redhat-operators catalog source.

       ```
       for catalog in $(ls oc-mirror-workspace/results-*/catalogSource* | grep -v spectrum-discover); do echo "Creating
       CatalogSource from file: $catalog"; echo "oc apply -f $catalog"; done
       ```

     - If a new TARGET_PATH value is used for the upgrade, then update the existing ImageContentSourcePolicy.

       ```
       cat << EOF > imagecontentsourcepolicy_dcs.yaml
       apiVersion: operator.openshift.io/v1alpha1
       kind: ImageContentSourcePolicy
       metadata:
         name: isf-dcs-icsp
       spec:
         repositoryDigestMirrors:
           - mirrors:
               - $LOCAL_ISF_REGISTRY/cpopen
             source: icr.io/cpopen
           - mirrors:
               - $LOCAL_ISF_REGISTRY/redhat
             source: registry.redhat.io/redhat
           - mirrors:
               - $LOCAL_ISF_REGISTRY/ubi8
             source: registry.redhat.io/ubi8
           - mirrors:
               - $LOCAL_ISF_REGISTRY/amq-streams
             source: registry.redhat.io/amq-streams
           - mirrors:
               - $LOCAL_ISF_REGISTRY/openshift4
             source: registry.redhat.io/openshift4
           - mirrors:
               - $LOCAL_ISF_REGISTRY/cp/ibm-spectrum-discover
             source: cp.icr.io/cp/ibm-spectrum-discover
           - mirrors:
               - $LOCAL_ISF_REGISTRY/db2u
             source: icr.io/db2u
       EOF
       oc apply -f imagecontentsourcepolicy_dcs.yaml
       ```

3. Fusion Data Foundation offline service upgrade:
   Offline upgrade of Data Foundation images that are deployed on OpenShift® Container Platform version 4.12, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, or 4.20.
   a. Complete steps ../sdsinstall/sds_fdf_mirror_images.html#sds_odf_mirror_images__step_hdp_5j5_fyb to ../sdsinstall/sds_fdf_mirror_images.html#sds_odf_mirror_images__step_rqc_vj5_fyb of Mirroring Fusion Data Foundation images topic.
   b. Before you upgrade IBM Fusion, from the Services page of the IBM Fusion user interface, disable Automatic updates for Data Foundation service.
   c. Go to Operators > Installed Operators > Fusion Data Foundation > Subscription, and check whether the Update approval is changed to Manually.
   d. Start the IBM Fusion version upgrade.
   e. Update the image digest ID after you upgrade the IBM Fusion as follows:
      i. Run the following command to get the catalog source image digest ID.

         ```
         skopeo inspect docker://<enterprise registry host:port>/<target-path>/cpopen/isf-data-foundation-catalog:<ocp
         version> | jq -r ".Digest"
         ```

         You need to record the image digest ID. It is used in deployment phase only.

      ii. Check whether the `data-foundation-service` `FusionServiceDefinition` CR is created.

         ```
         oc get fusionservicedefinitions.service.isf.ibm.com -n ibm-spectrum-fusion-ns data-foundation-service
         ```

      iii. Update the `imageDigest` in the `FusionServiceDefinition` `data-foundation-service`.

         ```
         skopeo inspect docker://<enterprise registry host:port>/<target-path>/cpopen/isf-data-foundation-catalog:<ocp
         version> | jq -r ".Digest"
         ```

      iv. Edit the `data-foundation-service` `.spec.onboarding.serviceOperatorSubscription.multiVersionCatSrcDetails.ocp412-t.imageDigest`.

         ```
         oc edit fusionservicedefinitions.service.isf.ibm.com -n ibm-spectrum-fusion-ns data-foundation-service
         ```

         Example of OpenShift Container Platform 4.12 output:

```
spec:
  hasRelatedDefinition: false
  onboarding:
...
    serviceOperatorSubscription:
      catalogSourceName: isf-data-foundation-catalog
      createCatalogSource: true
      globalCatalogSource: true
      isClusterWide: false
      multiVersionCatSrcDetails:
        ocp49:
          skipCatSrcCreation: true
        ocp410:
          skipCatSrcCreation: true
        ocp411:
          skipCatSrcCreation: true
        ocp412-t:
          displayName: Data Foundation Catalog
          imageDigest: sha256:ed94a66296d1a4fe047b0a79db0e8653e179a8a2a646b0c05e435762d852de73
          imageName: isf-data-foundation-catalog
          imageTag: v4.12
          publisher: IBM
          registryPath: icr.io/cpopen
          skipCatSrcCreation: false
```

f. Change Update approval to the original value in the IBM Fusion user interface.

4. Modify the image content source policy **isf-operator-index**. For each source defined in the image content source policy, add the new mirror that points to the new registry. If you want to mirror to the same enterprise registry as the previous version, then skip this step.
   See the following sample image content source policy:

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: isf-catalog-index
spec:
  repositoryDigestMirrors:
  # for scale
  - mirrors:
    - <Old ISF enterprise registry host>/<Old ISF target-path>
    - <Old ISF enterprise registry host:port>/<Old ISF target-path>
    - <New ISF enterprise registry host>/<New ISF target-path>
    - <New ISF enterprise registry host:port>/<New ISF target-path>
    source: cp.icr.io/cp/spectrum/scale
  - mirrors:
    - <Old ISF enterprise registry host>/<Old ISF target-path>
    - <Old ISF enterprise registry host:port>/<Old ISF target-path>
    - <New ISF enterprise registry host>/<New ISF target-path>
    - <New ISF enterprise registry host:port>/<New ISF target-path>
    source: icr.io/cpopen
  #for IBM Spectrum Fusion operator
  - mirrors:
    - <Old ISF enterprise registry host>/<Old ISF target-path>
    - <Old ISF enterprise registry host:port>/<Old ISF target-path>
    - <New ISF enterprise registry host>/<New ISF target-path>
    - <New ISF enterprise registry host:port>/<New ISF target-path>
    source: cp.icr.io/cp/isf-sds
  # for spp agent
  - mirrors:
    - <Old ISF enterprise registry host>/<Old ISF target-path>/sppc
    - <Old ISF enterprise registry host:port>/<Old ISF target-path>/sppc
    - <New ISF enterprise registry host>/<New ISF target-path>/sppc
    - <New ISF enterprise registry host:port>/<New ISF target-path>/sppc
    source: cp.icr.io/cp/sppc
  - mirrors:
    - <Old ISF enterprise registry host>/<Old ISF target-path>/sppc
    - <Old ISF enterprise registry host:port>/<Old ISF target-path>/sppc
    - <New ISF enterprise registry host>/<New ISF target-path>/sppc
    - <New ISF enterprise registry host:port>/<New ISF target-path>/sppc
    source: registry.redhat.io/amq7
  - mirrors:
    - <Old ISF enterprise registry host>/<Old ISF target-path>/sppc
    - <Old ISF enterprise registry host:port>/<Old ISF target-path>/sppc
    - <New ISF enterprise registry host>/<New ISF target-path>/sppc
    - <New ISF enterprise registry host:port>/<New ISF target-path>/sppc
    source: registry.redhat.io/oadp
  - mirrors:
    - <New ISF enterprise registry host>/<New ISF target-path>/sppc
    - <New ISF enterprise registry host:port>/<New ISF target-path>/sppc
    source: registry.redhat.io/amq-streams
```

Upgrading IBM Fusion

This topic outlines the steps to upgrade IBM Fusion management software, Fusion services, and Red Hat® OpenShift® Container Platform.

Before you begin

- If you installed the current IBM Fusion version by using online or your enterprise registry installation mode, then ensure that you do not change the mode during the upgrade to 2.12.0. To change the installation mode, reinstall IBM Fusion.
- Ensure all compute nodes are in ready state on OpenShift user interface.

- Download the logs that you collected by using IBM Fusion. The Collect logs user interface page gets deleted after the upgrade process completes so download the needed logs before you begin the upgrade.
- If you installed the earlier version of IBM Fusion by using your enterprise registry, then do these additional mirroring prerequisites steps. See Prerequisites for enterprise registry upgrade.
- The IBM Fusion operator upgrade can get stuck during upgrade due to a known Red Hat OLM issue. The issue is specific to Red Hat OpenShift Container Platform versions 4.16, 4.17, and 4.18. Resolve the problem before you proceed with the upgrade. For the resolution steps, see Hot fix page.
- For IBM Fusion cluster setup with automatic upgrade strategy from version 2.11.0 to 2.12.0, you need to manually approve the `installplan` generated for the upgrade on the respective cluster.

About this task

- Important: From IBM Fusion version 2.13.0, you can directly upgrade from 2.11.x to 2.13.0. For more information, see Enhanced upgrade experience.
- Upgrade paths:

Source version	Target version
2.9.0	2.9.1 > 2.10.0>2.11
2.9.1	2.10.0>2.11
2.10.0	2.11.0
2.11.0	2.12.0>2.12.1>2.12.2
2.12.0	2.12.1>2.12.2>2.13.0
2.12.1	2.12.2>2.130
2.12.2	2.13.0

Procedure

1. Log in to the OpenShift Container Platform management console as the cluster administrator.
2. Upgrade IBM Fusion:
   a. From the navigation menu, click Operators > Installed Operators.
   b. From the Installed Operators list, click IBM Storage Fusion operator.
   The Details tab opens by default.
   c. Go to Subscription tab.
   d. View the Subscription details section for the upgrade status.
   Note: If this is an offline setup, then update the image path in IBM Fusion catalog source with new catalog source image.
   e. If an upgrade is available for the operator, then click Approve to manually initiate the upgrade. If you do not agree to the upgrade, click Deny.
   If no new upgrade is available, then Upgrade status displays Up to date.
   Note: By default, the upgrade of the IBM Fusion is Manual.
   f. After the upgrade is successful, refresh your browser and clear your cache.
   g. Verify whether the IBM Fusion is in succeeded state and the version is 2.10.0 or 2.10.1. Also, in the Subscription tab, ensure that the upgrade status displays Up to date.
3. Upgrade IBM Fusion services.
   For the steps to upgrade services, see Upgrading IBM Fusion services.
4. Upgrade Red Hat OpenShift Container Platform from Red Hat OpenShift Container Platform console.
   Note: The Red Hat OpenShift Container Platform version must be upgraded only to the version supported by IBM Fusion. Do not upgrade or rollback your Red Hat OpenShift Container Platform version manually until the support is included in IBM Fusion.
   For the Red Hat OpenShift Container Platform upgrade procedure, see Understanding OpenShift Container Platform updates in *Red Hat OpenShift Container Platform* documentation.

- **Enhanced upgrade experience**
  This topic describes the process for upgrading IBM Fusion from version 2.11.x directly to 2.13.0, bypassing the intermediate 2.12.x releases, on supported configurations.

Enhanced upgrade experience

This topic describes the process for upgrading IBM Fusion from version 2.11.x directly to 2.13.0, bypassing the intermediate 2.12.x releases, on supported configurations.

If you are on IBM Fusion 2.11.x version using the `ibm-operator-catalog` catalog source and want to upgrade to 2.12.x without upgrading to 2.13.0, you must change your catalog source to use the IBM Fusion 2.12.x catalog source.

To upgrade IBM Fusion from 2.11.x to 2.13.0 directly, ensure that the following prerequisites are met:

- IBM Fusion version must be on 2.11.x.
- The OpenShift® version is either 4.18.x or 4.19.x.
- Fusion Data Foundation or Global Data Platform Remote Mount is supported only on OpenShift4.18.
- If using Content-Aware Storage 1.0.x, upgrade to at least Content-Aware Storage 1.0.7 before initiating the IBM Fusion HCI upgrade.

Table 1. Supported Enhanced Upgrade Topologies

Support matrix for enhanced upgrade experience from 2.11.x to 2.13.0					
Topology	**Fusion Data Foundation**	**Global Data Platform Remote Mount**	**Backup & Restore**	**IBM Data Cataloging**	**Content-Aware Storage**
OpenShift Container Platform 4.18	Supported	Supported	Supported	Supported	Supported
OpenShift Container Platform 4.19	Supported	Not supported	Supported	Supported	Supported

Upgrading Hosted Control Plane Fusion and its services

When IBM Fusion and services on the Hosted Control Plane cluster are installed using labels, they are automatically upgraded to the latest supported version available on the IBM Fusion hub.

- **Upgrading OpenShift Container Platform on Hosted Control Plane**
  Upgrade the OpenShift Container Platform version for Hosted Control Planes (HCP) clusters by following the Red Hat documentation and updating the NodePool custom resource to point to the mirrored release image for offline environments.

Upgrading OpenShift Container Platform on Hosted Control Plane

Upgrade the OpenShift Container Platform version for Hosted Control Planes (HCP) clusters by following the Red Hat documentation and updating the NodePool custom resource to point to the mirrored release image for offline environments.

To upgrade the OpenShift® Container Platform version for Hosted Control Plane clusters, see the Updating a control plane in a hosted cluster in *Red Hat OpenShift Container Platform* documentation.
Note: For offline Hosted Control Plane clusters:

- If you are upgrading OpenShift Container Platform in an offline Hosted Control Plane cluster, first mirror the images of the desired OpenShift Container Platform version that you want to upgrade to.
- In addition to the steps provided in the Red Hat documentation, make sure to update the `release-image` field in the `NodePool` custom resource (CR) to point to the mirrored OpenShift Container Platform release image.

Upgrading OpenShift Container Platform

Upgrade OpenShift in IBM Fusion. For offline upgrades of OpenShift Container Platform, follow the Red Hat steps. However, follow the guidelines and versions that are specified in this section.

Before you begin

Important: If you are upgrading to OpenShift Container Platform 4.21 for the Fusion Data Foundation service and you selected Enable automatic updates during installation, you must disable automatic upgrades if the Global Data Platform service is installed or configured. Before you proceed with the upgrade, review the supported migration scenarios.
Important: Check the compatibility of your underlying storage before you proceed with the OpenShift Container Platform upgrade. For more information about the version support, see Support matrix.

- Common prerequisites:
  - After every OpenShift Container Platform upgrade, check whether you have an installation of OpenShift virtualization operator. If it is available, then upgrade it. For more information, see Updating OpenShift Virtualization in *Red Hat OpenShift Container Platform* documentation.
  - If you are using an offline method, see the following *Updating clusters topics* based on your version of OpenShift Container Platform:
    - Red Hat OpenShift Container Platform 4.16 documentation
    - Red Hat OpenShift Container Platform 4.17 documentation
    - Red Hat OpenShift Container Platform 4.18 documentation
    - Red Hat OpenShift Container Platform 4.19 documentation
    - Red Hat OpenShift Container Platform 4.20 documentation
    - Red Hat OpenShift Container Platform 4.21 documentation
  - Verify whether all services and OpenShift Container Platform are healthy.
  - Both high speed switches must be available, functional, and healthy.
    Note: Review the supported and allowed upgrade path on OpenShift Container Platform upgrade graph tool. Do this to avoid the upgrade to a version that does not have any further upgrade paths due to unavailability of stable channels. For more information about upgrade version details, see Red Hat OpenShift Container Platform Update Graph.
- When the submariner auto upgrade fails or you observe any submariner issue during OpenShift Container Platform upgrade, contact IBM Support.
- Ensure that the appropriate eviction strategy is configured for the virtual machine before starting the OpenShift Container Platform upgrade. For more information, see Eviction strategies in *Red Hat OpenShift Container Platform* documentation.

About this task

In the following procedure, version 4.18 is used as an example. Replace it with your version of OpenShift Container Platform.

Procedure

1. Run the following command to change the channel to stable 4.18 or higher.

   ```
   oc patch clusterversion version --type merge -p '{"spec": {"channel": "stable-4.18"}}'
   ```

2. Upgrade OpenShift Container Platform to 4.18.18.
   Important: When you upgrade OpenShift Container Platform from version 4.20 to 4.21, run the following command to acknowledge that the Sigstore signatures are mirrored and your cluster is ready update:

   ```
   oc -n openshift-config patch configmap admin-acks --patch '{"data":{"ack-4.20-sigstore-in-4.21":"true"}}' --type=merge
   ```

3. Wait for upgrade to complete.

4. Verify whether all services and OpenShift Container Platform are healthy.

   OpenShift Container Platform console:
   Go to Operators > Installed operator in IBM Fusion namespace and check that the ISF operator shows in Succeed state.

   OC commands
   - Run the following command to check the ready state of the OpenShift Container Platform nodes:

     ```
     oc get nodes
     ```

   - Run the following command to check that all operators are available.

     ```
     oc get co
     ```

Upgrading IBM Fusion services

From the IBM Fusion user interface, upgrade the IBM Fusion services, namely Data Foundation, Global Data Platform, and IBM Data Cataloging. The Backup & Restore service auto upgrades based on availability and you can monitor the progress.

Before you begin

Important: It is recommended to upgrade the services to the latest version after a IBM Fusion upgrade to avoid compatibility issues between IBM Fusion and its installed services.

About this task

You can upgrade the services from the Services page or the Settings > Upgrades page of the IBM Fusion user interface.

When you upgrade from the Services page, you can initiate upgrade of individual services based on the availability of a new version. In the Upgrade page, you can initiate multiple non-disruptive and rolling upgrades of service in parallel. For both non disruptive and rolling upgrades, you can view the status of the prechecks and monitor the progress of the upgrade. In the Services page, click View upgrade details of a ongoing service upgrade and in the Upgrade page, you can click the service tile of an ongoing upgrade to monitor the pre-check results and the progress of the upgrade in a slide out pane.

In addition, in the Upgrade page, you can do rolling disruptive upgrade where the upgrade mandates a node restart. To minimize the impact on the workloads, these restarts happen one at a time.

View the availability of an upgrade
You can view the availability of an upgrade in the following user interface pages:

- In OpenShift® Container Platform web management console, go to Overview page and check the Activity section.
- In the IBM Fusion user interface:
  - Click the bell icon and view notifications in System notifications.
  - Check the notification banner on the Services page.
  - Check the notification banner on the Settings > Upgrade page.

Pre-checks
When you initiate upgrade, the Upgrade <service name> window gets displayed. The IBM Fusion runs common pre-checks that impact the upgrade of the service. For more information about the prechecks and its coverage, see the user interface. If all the checks pass or there are some conditions in warning state, you can proceed with the upgrade. However, it is highly recommended to fix the warnings to prevent any potential impacts on the system during the upgrade process. Immediately address any blocker errors encountered during prechecks, as they require fixing before proceeding with the upgrade. After the completion of all pre-checks, a summary of total number of issues found gets displayed. Wait for a moment to see the precheck status in the IBM Fusion user interface.
You can click on the corresponding BMYxxxxxx code to know more about how to fix the issue. In the Upgrade precheck report window, if you find error resources in the Resources section, you can click the resource link to know more details about it.

- **Migrating Global Data Platform**
  Migrate the Global Data Platform service when installing or upgrading Fusion Data Foundation 4.21 or later.
- **Upgrading Global Data Platform**
  Upgrade Global Data Platform service.
- **Upgrading Fusion Data Foundation**
  Steps to upgrade the Fusion Data Foundation service.
- **Upgrade Data cataloging**
  Steps to upgrade IBM Data Cataloging service.
- **Upgrading Backup & Restore service**
  After a IBM Fusion operator upgrade, if a Backup & Restore upgrade is available, IBM Fusion automatically initiates prechecks and then proceeds with the upgrade.
- **Upgrading Content-Aware Storage (CAS)**
  Steps to upgrade Content-Aware Storage (CAS) service.

Migrating Global Data Platform

Migrate the Global Data Platform service when installing or upgrading Fusion Data Foundation 4.21 or later.

Before you begin

Ensure that the following conditions are met before starting the migration:

- No backup or restore jobs are running.

- The IBM Storage Scale cluster is healthy:
  - Run the `mmhealth` command.
  - Verify that all components report a healthy status.
- No MachineConfigPool (MCP) rollout is in progress.
- Global Data Platform version is 6.0.0.2. If the current version is 6.0.0.1, upgrade it to 6.0.0.2 before proceeding. For instructions, see Upgrading Global Data Platform.

About this task

Starting with Red Hat® OpenShift® Container Platform 4.21 and later, and with Fusion Data Foundation 4.21 and later, the Global Data Platform service is integrated into Fusion Data Foundation.

In earlier releases, Global Data Platform was bundled separately with IBM Fusion. From Fusion Data Foundation 4.21 onward:

- No separate upgrade option is available for version 6.0.1.0 and later. Global Data Platform upgrades and the migration of IBM Storage Scale remote mount (CNSA) from version 6.0.0.2 to 6.0.1.0 or later are performed automatically in the background during Fusion Data Foundation installation or upgrade.
- Each Fusion Data Foundation release includes an updated IBM Storage Scale build, which is applied automatically.
- Future updates to IBM Storage Scale are delivered through Fusion Data Foundation upgrades.

The migration workflow depends on your deployment scenario.

Procedure

Scenario 1: Only Global Data Platform is installed

If Global Data Platform 6.0.0.2 is installed and Fusion Data Foundation is not installed:

- Install Fusion Data Foundation 4.21 or later. For instructions, see Installing Fusion Data Foundation.

During installation:

- The system automatically triggers the Global Data Platform migration.
- Global Data Platform is upgraded to version 6.0.1.0 or later as part of the process.

Scenario 2: Both Fusion Data Foundation and Global Data Platform are installed

If Fusion Data Foundation 4.20 or earlier and Global Data Platform 6.0.0.2 are already installed:

1. Disable the Fusion Data Foundation automatic upgrade to prevent unintended upgrade behavior.
2. Manually trigger the Fusion Data Foundation upgrade from version 4.20 to 4.21 or later. For instructions, see Upgrading Fusion Data Foundation.

During the upgrade:

- Global Data Platform migration runs and completes automatically.
- No separate action is required for migration.

Results

After the Fusion Data Foundation installation or upgrade and the Global Data Platform migration are complete:

- The Global Data Platform tile is removed from the IBM Fusion user interface.
- The Remote Filesystems page is no longer available in the IBM Fusion user interface.
- All IBM Storage Scale pods run in a single namespace, `ibm-spectrum-scale`.
- In the Red Hat OpenShift Container Platform web console, navigate to Storage > External Systems to view version details and remote cluster information.

What to do next

To manage and configure Global Data Platform after migration, go to Data Foundation > External Storage in the Red Hat OpenShift Container Platform web console. For more information, see Configuring with IBM Storage Scale.

Upgrading Global Data Platform

Upgrade Global Data Platform service.

Before you begin

Important:
Check your Red Hat® OpenShift® Container Platform version:

- **Version 4.20 or earlier**: Continue with this procedure
- **Version 4.21 or later**: see Migrating Global Data Platform

- Ensure no Backup & Restore jobs are running.
- Run the `mmhealth` command to check the scale common health status of the storage scale cluster.

About this task

You can upgrade to Global Data Platform from the Services page or Settings > Upgrades page.

For more information about the storage version support, see Support matrix.

Procedure

1. In the Installed section, click Upgrade in Global data platform. Alternatively, click Upgrade in the ellipsis menu of Global data platform.
   Only if an upgrade is available for Global data platform, the Upgrade option is displayed. Alternatively, you can go to the Settings > Upgrades page and view the upgrade status and in progress upgrades. The Global Data Platform goes into rolling upgrade. If you can run service upgrades in parallel in a non-disruptive manner, click Upgrade (<number of upgrades available>). If there are any errors found in the automatic prechecks, then click the service tile for more details about the error. The Upgrade global data platform window gets displayed. It displays the upgrade version and approximate time to complete the upgrade.
2. In Upgrade global data platform window, click Upgrade.
   The upgrade starts and you can see the upgrade status. The upgrade starts and you can see the upgrade status. The IBM Fusion runs common pre-checks during upgrade. You can monitor each step, with a clear understanding of what is in progress. If anything gets stuck, a new guided workflow shows you what needs your attention and guides you with the resolution. After the completion of upgrade, a success notification gets displayed. For more information about the pre-checks, see Pre-checks in *About this task* section.
   Note: Wait for a minute or more to see the upgrade status in the IBM Fusion user interface.
   If you see the "**`Scale is in critical state`**" error message on the screen, ignore it as IBM Storage Scale goes back to healthy state after the upgrade completes successfully.
   If you see "**`Service upgrade has failed`**" message, then wait for the upgrade to retry automatically. If issues do not exist in your setup, then upgrade completes successfully.
3. Run the following commands to delete the resources that are deprecated in IBM Storage Scale. For more information, see Remove deprecated resources.

   ```
   oc delete ClusterRoleBinding ibm-spectrum-scale-dns
      oc delete ClusterRole ibm-spectrum-scale-dns
   ```

Upgrading Fusion Data Foundation

Steps to upgrade the Fusion Data Foundation service.

Before you begin

- Important: If you upgrade to Fusion Data Foundation 4.21 or later and the Global Data Platform service is installed or configured, the service is automatically migrated to version 6.0.1.0. If automatic updates are enabled, disable automatic upgrades before proceeding. Review the supported migration scenarios before you proceed with the upgrade.
- For the Fusion Data Foundation service upgrade, follow this upgrade sequence: 4.16.x > 4.17.x> 4.18.x> 4.19.x> 4.20.x.
- If you plan to do a offline installation in IBM Fusion 2.13.0, mirror the Fusion Data Foundation images to your enterprise registry. For the procedure and images to download, see Mirroring Fusion Data Foundation images.

About this task

Data Foundation service can be set to auto update or manual update. You can set this value during the installation and change after installation. If Enable automatic updates is selected during installation, you can ignore the rest of the upgrade steps.
If Disable automatic updates is set and a new Data Foundation service version is available, then a message prompts you to upgrade. For example, `Service upgrade available, Version 4.20 is available for Data Foundation service`.
Note: During a IBM Fusion upgrade on a hosted cluster, you may briefly see the Fusion Data Foundation service status as `Upgrade Available` or `Upgrading`. If you upgraded OpenShift Container Platform after the installation of Fusion Data Foundation service, then this behavior is expected, and the Fusion Data Foundation client reports as healthy again after a short time.
Do the following steps to manually upgrade the service:

Procedure

1. Log in to IBM Fusion user interface.
2. Click Services menu to go to the Services page.
   Alternatively, you can go to the Settings > Upgrades page and view the upgrade status and in progress upgrades. If you can run service upgrades in parallel in a non-disruptive manner, click Upgrade (<number of upgrades available>). If there are any errors found in the automatic prechecks, then click the service tile for more details about the error.
3. In the Installed section, click Upgrade link for the Data Foundation service. Alternatively, from the ⋮ (ellipsis overflow menu) of Data Foundation, click Upgrade.
   You can see the Upgrade option only when an upgrade is available for the service.
   The upgrade starts and you can see the upgrade status. The IBM Fusion runs common pre-checks during upgrade. You can monitor each step, with a clear understanding of what is in progress. If anything gets stuck, a new guided workflow shows you what needs your attention and guides you with the resolution.
4. Wait for a minute or more to see the upgrade status in the IBM Fusion user interface.
   After the upgrade completes, a success notification gets displayed.
   Note: If you see **`Service upgrade has failed`** message, then wait for the upgrade to retry automatically. If issues do not exist in your setup, then upgrade completes successfully.
5. To change the auto update setting after installation, do the following steps:
   a. In the Installed section of the Services page, click the ellipsis menu of Data Foundation service.
   b. Click Enable automatic updates or Disable automatic updates option. If you enable automatic updates, then Disable automatic updates option is available in the ellipsis overflow menu.

- **Upgrading Red Hat OpenShift Data Foundation to IBM Storage Fusion Data Foundation**
  Steps to upgrade Red Hat Data Foundation to IBM Fusion Data Foundation.

Upgrading Red Hat OpenShift Data Foundation to IBM Storage Fusion Data Foundation

Steps to upgrade Red Hat® Data Foundation to IBM Fusion Data Foundation.

Before you begin

- Important: If OpenShift® Data Foundation 4.20 is installed and the Global Data Platform is deployed on IBM Fusion, upgrade to Fusion Data Foundation. Starting with Red Hat OpenShift Container Platform4.21 or later, IBM Storage Scale is integrated with Fusion Data Foundation 4.21 or later. Upgrade from OpenShift Data Foundation to Fusion Data Foundation to use Global Data Platform capabilities.
- Ensure that the OpenShift Data Foundation version is the same as or lower than the planned upgrade version of Fusion Data Foundation.

  To upgrade to a specific version of Fusion Data Foundation, your OpenShift Data Foundation version should be the same or lower. For example, to upgrade to version 4.18.0, your current version should be 4.18.0 or lower, such as 4.17.x. Similarly, if you are upgrading to 4.18.2, a current version like 4.18.1 is compatible because its lower than 4.18.2. This rule applies to all version levels, including major, minor, and patch versions.

- Ensure that the OpenShift Data Foundation cluster is healthy and data is resilient. You can get the details from IBM Fusion user interface or OpenShift Container Platform console.
- Ensure that all OpenShift Data Foundation Pods, including the operator pods, are in Running state in the openshift-storage namespace.
- Ensure that you have sufficient time to complete the OpenShift Data Foundation update process. The update time varies depending on the number of OSDs that run in the cluster.
- Ensure IBM Fusion Data Foundation catalogsource exists:
  1. Verify that the **isf-data-foundation-catalog** catalog exists. The IBM Fusion Data Foundation catalog **isf-data-foundation-catalog** gets created automatically in **openshift-marketplace** ns.

     ```
     oc get catalogsources.operators.coreos.com -n openshift-marketplace
     ```

     Sample output:

     ```
     NAME                          DISPLAY                            TYPE   PUBLISHER   AGE
     isf-data-foundation-catalog   IBM Storage Fusion Data Foundation   grpc   IBM         66m
     ```

  2. If the **isf-data-foundation-catalog** catalog does not exist, create it as follows:
     a. Create a **CatalogSource** object that references the catalog image for **isf-data-foundation-catalog**.
     b. Ensure the **CatalogSource** object uses the correct registry image.
     For example:

     ```
     apiVersion: operators.coreos.com/v1alpha1
     kind: CatalogSource
     metadata:
       name: isf-data-foundation-catalog
       namespace: openshift-marketplace
     spec:
       displayName: isf-data-foundation-catalog
       publisher: IBM
       sourceType: grpc
       image: icr.io/cpopen/isf-data-foundation-catalog:v4.19
       updateStrategy:
         registryPoll:
           interval: 15m
     ```

     c. Create a **CatalogSource** using this file by running the following command:

     ```
     oc apply -f isf-data-foundation-catalogsource.yaml
     ```

Procedure

1. Change update approval strategy to Manual.
   a. Change update approval strategy to Manual in OpenShift Data Foundation subscription
2. Modify OpenShift Data Foundation subscription catalog source name.
   a. Update the odf-operator subscription.

   ```
   oc patch -n openshift-storage subscriptions.operators.coreos.com/odf-operator --patch '{"spec":{"source":"isf-data-foundation-catalog"}}' --type=merge
   ```

   b. Run the following command to get all the subscriptions under **openshift-storage** namespace:

   ```
   oc get subscriptions.operators.coreos.com -n openshift-storage
   ```

   Sample output:

   ```
   NAME                                                                PACKAGE                SOURCE
   cephcsi-operator-stable-4.17-redhat-operators-openshift-marketplace   cephcsi-operator       redhat-operators
   ...
   ...
   ```

   c. Update the **Catalogsources** for all subscriptions:
   For example:

   ```
   oc patch -n openshift-storage $(oc get subs -n openshift-storage -l operators.coreos.com/cephcsi-operator.openshift-storage= -o name) --patch '{"spec":{"source":"isf-data-foundation-catalog"}}' --type=merge

   oc patch -n openshift-storage $(oc get subs -n openshift-storage -l operators.coreos.com/ocs-operator.openshift-
   ```

```
storage= -o name) --patch '{"spec":{"source":"isf-data-foundation-catalog"}}' --type=merge

oc patch -n openshift-storage $(oc get subs -n openshift-storage -l operators.coreos.com/mcg-operator.openshift-
storage= -o name) --patch '{"spec":{"source":"isf-data-foundation-catalog"}}' --type=merge

oc patch -n openshift-storage $(oc get subs -n openshift-storage -l operators.coreos.com/odf-csi-addons-
operator.openshift-storage= -o name) --patch '{"spec":{"source":"isf-data-foundation-catalog"}}' --type=merge
```

3. Verify the OpenShift Data Foundation subscription's catalogsource name is changed.
   a. Ensure all the catalogsource name is updated.

   ```
   oc get subscriptions.operators.coreos.com -n openshift-storage
   ```

   Sample output:

   ```
   NAME                                                                PACKAGE                  SOURCE
   mcg-operator-stable-4.17-redhat-operators-openshift-marketplace     mcg-operator             isf-data-foundatio
   ocs-operator-stable-4.17-redhat-operators-openshift-marketplace     ocs-operator             isf-data-foundatio
   odf-csi-addons-operator-stable-4.17-redhat-operators-openshift-marketplace   odf-csi-addons-operator   isf-data-foundatio
   odf-operator
   ```

4. Review the installpan and click Approve, and then wait until all the IBM Fusion Data Foundation pods are upgraded.
5. Verify the operator status.
   a. Check the Version of **Fusion Data Foundation** and the operator status
      i. Go to Operators > Installed Operators and select the openshift-storage project.
      ii. When the upgrade completes, the version updates to a new version number for **Fusion Data Foundation** and status changes to **Succeeded** with a green tick.
6. Verify **Fusion Data Foundation** cluster is healthy and data is resilient in the IBM Fusion user interface > Data Foundation page. Alternatively, go to Storage > Data Foundation > Storage Systems tab and then click the storage system name. Check for the green tick on the status card of **Overview- Block and File** and **Object** tabs. Green tick indicates that the storage cluster, object service, and data resiliency are healthy.
7. Verify all the CSVs under **openshift-storage** are **Succeeded**.

   ```
   oc get csv -n openshift-storage
   ```

   Sample output:

   ```
   NAME                              DISPLAY                          VERSION   REPLACES                                PHASE
   mcg-operator.v4.17                NooBaa Operator                  4.17    mcg-operator.v4.17-rhodf                Succeeded
   ocs-operator.v4.17                Container Storage                4.17   ocs-operator.v4.17-rhodf                Succeeded
   odf-csi-addons-operator.v4.17   CSI Addons                          4.17    odf-csi-addons-operator.v4.17-rhodf    Succeeded
   odf-operator.v4.17                IBM Storage Fusion Data Foundation   4.17    odf-operator.v4.17-rhodf             Succeeded
   ```

   Note: If any of the verification steps fail, contact IBM Support.
8. Set the update strategy approval back to Automatic to deploy the latest version as soon as its available.

Upgrade Data cataloging

Steps to upgrade IBM Data Cataloging service.

About this task

Sometimes, the IBM Data Cataloging service upgrade may not be available after you upgrade IBM Fusion. To know more about such issues and its resolutions, see IBM Data Cataloging service installation and upgrade issues.

Procedure

Upgrade **Data Cataloging** service as follows:

a. In the Installed section, click Upgrade in the ellipsis overflow menu of Data Cataloging.
   Alternatively, you can go to the Settings > Upgrades page and view the upgrade status and in progress upgrades. If you can run service upgrades in parallel in a non-disruptive manner, click Upgrade (<number of upgrades available>). If there are any errors found in the automatic prechecks, then click the service tile for more details about the error.
b. In the Upgrade service window, click **Upgrade**.
   The upgrade starts and you can see the upgrade status. The IBM Fusion runs common pre-checks during upgrade. You can monitor each step, with a clear understanding of what is in progress. If anything gets stuck, a new guided workflow shows you what needs your attention and guides you with the resolution. After the completion of upgrade, a success notification gets displayed.
   Note: Wait for a minute or more to see the upgrade status in the IBM Fusion user interface.

Upgrading Backup & Restore service

After a IBM Fusion operator upgrade, if a Backup & Restore upgrade is available, IBM Fusion automatically initiates prechecks and then proceeds with the upgrade.

Before you upgrade

When the Backup & Restore service is upgraded, any custom plugins added to the **DataProtectionApplication** (DPA) are overwritten. If you or a service such as Cloud Pak for Data or Cloud Pak for Integration has added custom entries to **.spec.velero.customPlugins**, those entries are erased when the new DPA is applied during the upgrade.

Important: Before you start the upgrade, record the current custom plugins in the DPA so that you can reapply them after the upgrade is complete. Custom `.spec.velero.customPlugins` entries are lost during upgrade and must be manually restored. The `isf-kubevirt` plugin is registered automatically by IBM Fusion and does not need to be reapplied.

1. Before upgrading, save your current DPA custom plugins:

   ```
   oc get dataprotectionapplication -n ibm-spectrum-fusion-ns \
     -o jsonpath='{.items[0].spec.velero.customPlugins}' | python3 -m json.tool
   ```

   Record all entries except `isf-kubevirt`.

2. Proceed with the IBM Fusion upgrade.
3. After the upgrade completes, reapply each saved custom plugin entry to the DPA:

   ```
   oc patch dataprotectionapplication <dpa-name> -n ibm-spectrum-fusion-ns \
     --type=json \
     -p='[{"op":"add","path":"/spec/velero/customPlugins/-","value":{"image":"<plugin-image>","name":"<plugin-name>"}}]'
   ```

   Repeat for each custom plugin that was present before the upgrade.

Important: When you upgrade to version 2.13.0, the Backup & Restore database is upgraded and the data is migrated to the new version of MongoDB. Ensure that the Backup & Restore storage class contains sufficient available capacity. The upgrade process requires an extra 60 GiB of storage temporarily to migrate the Backup & Restore database to the new version. After the migration is complete, the old Persistent Volume Claims (PVCs) are deleted, releasing the extra storage.
If you plan to do an offline installation in IBM Fusion 2.13.0, mirror the Backup & Restore 2.13.0 images to your enterprise registry for the automatic upgrade to initiate. For the procedure and images to download, see Mirroring Backup & Restore images.

To view the status of the upgrade, go to Settings > Upgrade or the Services page.

Upgrading Content-Aware Storage (CAS)

Steps to upgrade Content-Aware Storage (CAS) service.

Before you begin

Open the IBM Fusion user interface, go to the Services tab and check the current version of CAS in the Installed section.
Note: You cannot skip version when upgrading CAS to the latest version. For example, if CAS v1.0.4 is installed as your current version, upgrading to v1.1.0 requires an upgrade path of 1.0.4 > 1.0.5 > 1.0.6 > 1.0.7 > 1.1.0.

Procedure

1. Open the IBM Fusion user interface.
2. Go to the Services tab. In the Installed section, click Upgrade service in the ellipsis overflow menu of Content-Aware Storage.
   Alternatively, you can go to the Settings > Upgrades page, and view the upgrade status and in progress upgrades. If you can run service upgrades in parallel in a non-disruptive manner, click Upgrade (<number of upgrades available>).
3. In the Upgrade service window, click Upgrade.
   The upgrade starts and you can see the upgrade status. After the completion of upgrade, a success notification is displayed.
   Note: Wait for a minute or more to see the upgrade status in the Services tab of the IBM Fusion user interface. You can also find the latest version in the Services tab when the upgrade completes.

Upgrade considerations for CAS v1.0.6 and earlier to v1.1.0
If you are upgrading from CAS v1.0.6 or earlier, start by upgrading to CAS v1.0.7 through the IBM Fusion user interface. After IBM Fusion initiates the upgrade, you must manually approve the process in OpenShift® before it can continue to CAS v1.1.0 by following these steps:

1. Manual approval that is required after CAS v1.0.7
   After Fusion completes the upgrade to CAS v1.0.7, the next upgrade step (from v1.0.7 to v1.1.0) does not start automatically.

   Before you continue, verify that the CAS subscription in OpenShift is configured as follows:

   - **Update approval: Manual**

   OpenShift path:

   Operators > Installed Operators > Project: ibm-cas > Content Aware Storage > Subscription

2. Verify whether CAS v1.0.7 is installed correctly
   Before you approve the upgrade to version 1.1.0, verify that the environment is ready. To confirm Kafka readiness for the next upgrade step, ensure that Kafka is running in KRaft mode.

   ```
   oc get kafka -n ibm-cas -o yaml | yq '.items[0].status.kafkaMetadataState'
   ```

   Expected output:

   ```
   KRaft
   ```

   When the value is `KRaft`, you can approve the next upgrade safely.

3. Approve upgrade to CAS v1.1.0
   a. After confirming readiness, go to Operators > Installed Operators and select the project as ibm-cas.
   b. Open Content Aware Storage, go to the Subscription tab, and click 1 requires approval.
   c. Select Preview install plan.
   d. Click Approve.

   The installation process of CAS v1.1.0 is triggered.

While the upgrade is waiting for manual approval in OpenShift, IBM Fusion displays the service status as Upgrading. This behavior is expected. IBM Fusion remains in this state until you approve the upgrade to version 1.1.0 in the OpenShift UI. After approval and installation are complete, IBM Fusion updates the version and marks the service as Healthy.

Troubleshooting and serviceability

The steps to troubleshoot an issue vary depending on the problem. To help make relevant information available to you as quickly as possible, the log files and other tools for troubleshooting problems are consolidated.

- **Contacting IBM Support Center**
  The IBM® Support Center is available for various types of IBM software problems that IBM Fusion customers might encounter.
- **How to provide feedback**
  Customers can submit feedback on the IBM Documentation platform and on product documentation using the "Was this topic helpful?" rating on every page.
- **Serviceability in IBM Fusion**
  Monitor and troubleshoot IBM Fusion by using events, logs, and diagnostic tools.
- **Events and error codes message references in IBM Fusion**
  This reference information provides a consolidated view of all the events or error messages that can be displayed when the IBM Fusion system might results in an event or an error.
- **Troubleshooting issues in IBM Fusion**
  This topic covers the most common issues and their workarounds, which you might encounter while you work with the IBM Fusion software.

Contacting IBM Support Center

The IBM® Support Center is available for various types of IBM software problems that IBM Fusion customers might encounter.

Before you begin

Points to note when you contact the IBM Support Center:

1. For IBM Fusion, a serial number is not required.
2. Based on your issue, collect appropriate log files and diagnostic data.
3. IBM Support provides a time period within which IBM representative returns your call. Be sure that the person that you identified as your contact can be reached in the phone number.
4. An online case gets created to track the problem you are reporting. Note down the number for future reference and communication. If you need to make subsequent calls to discuss the problem, use the number to identify the problem.

About this task

If you enabled Call Home, then it connects your system to service representatives who can monitor issues and respond to problems efficiently and quickly to keep your system up and running. Enabling Call Home significantly improves your IBM Support experience. For more information about Call Home and Service level in Call Home, see Enabling Call Home.

If you have Software Maintenance service contract, then use this procedure to open a case on IBM Support site.

The following table lists alternative ways of contacting IBM Support Center:

Your location	Method of contacting the IBM Support Center
In the United States	Call **1-800-IBM-SERV** for support.
Outside the United States	Contact your local IBM Support Center or see the Directory of worldwide contacts (www.ibm.com/planetwide).

If you do not have an IBM Software Maintenance service contract, then contact your IBM sales representative to find out how to proceed. For non-IBM failures, follow the problem-reporting procedures provided with that product.

Procedure

1. Log in to IBM support page by using your IBMid and click Continue.
2. In Let's troubleshoot, click Open a case tile.
3. In the Open a case page, enter the following details:
   a. Select a value for the Type of support that you need.
   b. Enter the Case title to describe your case. The maximum length of the case title is 255 characters.
   c. Enter `IBM` in Product manufacturer.
   d. Enter `Fusion` for Product.
   e. Enter Product version.
   f. Select Service Type.
   g. Check Address where product is located.
   h. Select Severity.
   i. Enter System down (yes or no).
   j. Enter Failing Component (Software or Hardware)
   k. Enter Account name and Case description.
   l. Enter Case contact phone number.
   m. You can Add team members based on the need.

   Note: If a diagnostic file is required for this case, you can upload it after you submit this case.
4. Go through the terms and conditions and click Submit case.

How to provide feedback

Customers can submit feedback on the IBM Documentation platform and on product documentation using the "Was this topic helpful?" rating on every page.

IBM customers can provide feedback on two levels: feedback on the the platform (IBM Documentation site) and feedback on your product's documentation.

How to provide feedback on IBM Documentation platform

To leave feedback about the IBM Documentation platform (website), use the "Was this topic helpful?" at the top of this page. When you rate the content with a thumbs up or thumbs down, a dialog appears where you can leave optional comments.
We welcome any feedback that you have. You will not receive a direct response to your feedback, but your comments help us improve your IBM Documentation experience.

How to provide feedback on your product's documentation

IBM Documentation provides a single standardized process to leave feedback on your product using "Was this topic helpful?", located at the top of every page. When you rate the content with a thumbs up or thumbs down, a dialog appears where you can leave optional comments. IBM Documentation automatically captures both the product and the URL of the page you are leaving your feedback on.
We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. You will not receive a direct response to your feedback, but your comments help us improve our product documentation.
If you require support and a response to technical questions, visit IBM's technical support site to open a defect.

Serviceability in IBM Fusion

Monitor and troubleshoot IBM Fusion by using events, logs, and diagnostic tools.

Monitoring events

To understand Events page of IBM Fusion, see Analyzing events in IBM Fusion.

Collecting logs

To create component-wise logs and download them for diagnosis, see Collecting logs in IBM Fusion.

- **Analyzing events in IBM Fusion**
  IBM Fusion events are Kubernetes native events that include filter-specific labels.
- **Collecting logs in IBM Fusion**
  You can collect logs related to storage or backup components to help with system issue diagnosis.
- **Benefits of enabling Call Home**
  Configure and enable the call home feature as it improves the user experience when IBM Fusion is used.
- **Configuring License Service Reporter for IBM Fusion**
  Configure License Service Reporter to obtain a unified view of IBM Fusion licensing metrics. License Service Reporter supports this feature and can be extended to IBM Fusion. In addition to the IBM Fusion user interface, it provides an additional view of licensing metrics, offering deeper insights. It can also be configured across multiple clusters, enabling centralized monitoring and reporting of license usage in distributed environments.

Analyzing events in IBM Fusion

IBM Fusion events are Kubernetes native events that include filter-specific labels.

The following component-based events and labels are available for IBM Fusion:

Table 1. Component events and their labels

Component	Label
Installation component events	`isf.ibm.com/installer-component=spp`
Application events	`isf.ibm.com/application-name=<name of the application>`
Assign policy to application events	`isf.ibm.com/application-name=<name of the application>, isf.ibm.com/backuppolicy-name=<Backup policy name>`
Backup storage location events	Nil
Backup events for daily backup	`isf.ibm.com/application-name=<name of the application>, isf.ibm.com/backuppolicy-name=<Backup policy name>`
Restore application from backup	Nil
Storage events for scale	`isf.ibm.com/storage-type=scale`

You can view these events from IBM Fusion user interface, Red Hat® OpenShift® Container Platform console, or Kubernetes oc commands.

Steps to view events from IBM Fusion user interface

Events are listed as table rows with Severity, Category, Time, Description, Support ticket, and Resource columns. By default, all critical, warning, and informational events are displayed in the list. You can sort the columns Severity, Category, and Support ticket.

1. Log in to IBM Fusion user interface.
2. Click Events menu.
3. In the Events page, enter keywords in Search text to search for records. Also, filter records based on Severity and Category.
   The events are listed with the following headers by default:
   - Severity - Type of event. It can be Critical, Warning, or Informational.

- Category - Category of the event. For example, Backup and restore, Storage, and Other.
- Time - Date and time of the occurrence of the event
- Description - Description of the event.
- Support ticket - Indicates the ticket number associated with the event.
- Resource - Indicates the policy associated with the event.

You can use the downloads icon to download all events that have registered in the last 48 hours.

You can use the settings icon to select or clear the column headings from the table display. If you want to reset to default column headings, then click Reset to default.

You can also decide the number rows to display in a view from the Items per page drop-down list. Use the arrow keys to jump between the list of pages or use the drop-down list next to the arrow keys to jump to a specific topic.

The events exist in the system for the following time duration:

- Information events lasts for 3 to 4 hours
- Warning events lasts for 7 days
- Critical events lasts for 14 days

If any existing **`Warning`** or **`Critical`** event is known to be resolved, it can be marked as fixed from the IBM Fusion events page.

For a list of IBM Fusion events and error messages, see Events and error codes message references in IBM Fusion.

Steps to view events from Red Hat OpenShift Container Platform console

1. Log in to OpenShift Container Platform console.
2. Go to Overview > Events and view events.
3. In the Events page, you can filter based on resources and type. In addition, You can also type the name or message to filter.

Important: Also, you enable notifications of IBM Fusion events using OpenShift Container Platform **`Alertmanager`**. For steps, see IBM Fusion event alerts.

Command line or command prompt

You can run oc commands to view events:

```
oc get events -l isf.ibm.com/fusion
```

To display all IBM Fusion Kubernetes events, include the label **`isf.ibm.com/fusion`**. For specific component, use the appropriate label. For example, if you want to view all application events with name bookstore, then run the following oc command:

```
oc get events -l isf.ibm.com/application-name=bookstore
```

- **IBM Fusion event alerts**
  Use these instructions to enable notifications of IBM Fusion events using OpenShift Container Platform **`Alertmanager`**.

IBM Fusion event alerts

Use these instructions to enable notifications of IBM Fusion events using OpenShift® Container Platform **`Alertmanager`**.

About this task

IBM Fusion supports notifications for **`CRITICAL`** and **`WARNING`** events, which are visible in the IBM Fusion user interface. By configuring supported receivers in the OpenShift Container Platform, notifications can be routed through preferred channels such as Slack, email, and webhooks. This setup simplifies incident management by delivering alerts directly to configured notification channels. Additionally, multiple IBM Fusion clusters can be configured with similar settings to centralize event notifications in a single location.

Procedure

1. Log in to the OpenShift Container Platform web console.
2. Go to Administration > Cluster Settings.
3. In the Configuration tab, select the Alertmanager.
4. In the Alertmanager, click Create Receiver.
   The Create Receiver page opens.

5. Enter the Receiver name and select the channel that you prefer from the Receiver type drop-down list.
   For example, if you select the Slack as a preferred channel. Ensure that you must update the following details.
   a. Enter the Slack API URL.
   b. Enter the Channel name to send the notifications.
   c. Use Show advanced configuration drop-down to customize the Icon, Username, Title, and Text.
   Important: You can use the following labels and annotations to customize your message:

   ```
   Severity: {{ .CommonLabels.severity }}
   Hardware Address: {{ .CommonLabels.hwAddr }}
   Event Message: {{ .CommonLabels.alertMessage }}
   Alert Summary: {{ .CommonAnnotations.summary }}
   Alert Description: {{ .CommonAnnotations.description }}
   ```

   - Cluster - Add the cluster name manually.
   - Severity - **`Critical`** or **`Warning`**.

- Address - Hardware address generating the event (cannot be present for software).
- Event message - Description of the event.
- Alert summary - Static string of `IBM Fusion Event Alert`.
- Alert description - Static string `Alert sent from IBM Fusion Event Manager`.

d. Enter the label details in the Routing labels.
Important:
- Ensure that you enter `source = ISFEventManager` label in the Routing labels.
- Do not mark Send resolved alerts to this receiver? to true.

Optionally, you can also add `severity = CRITICAL` or `severity = WARNING` labels.

6. Click Create.

What to do next

After the receiver is configured, you start receiving notifications for IBM Fusion events based on the routing label specified during setup.

By default, when the routing label source is set to `ISFEventManager`, the receiver is notified of all events with severity levels `WARNING` and `CRITICAL`. This setup ensures that important alerts are delivered promptly without needing extra filtering.

To streamline monitoring across multiple racks, you can apply the same receiver configuration to each rack. This enables centralized alerting, allowing all events to be routed to a common notification channel for unified visibility and management.
Note: The cluster name is not automatically detected, so it is recommended to manually specify the cluster name in the receiver configuration to ensure accurate identification and traceability of events.

Collecting logs in IBM Fusion

You can collect logs related to storage or backup components to help with system issue diagnosis.

Procedure

1. From the title bar, click the help icon and select Support logs.
2. In the Support logs page, click Create.
The Create a log package page gets displayed.
3. In the Create a log package page, select one or more services to create the log package. The log package contains the folders, namespaces, and relevant files. For more guidance on the log package, see Log package.

Software and services
Software and services logs contains the information about the operation, status, and potential issues of the services such as Global Data Platform, IBM Data Cataloging, Backup & Restore, Data Foundation, and CAS in the IBM Fusion.

Global Data Platform
Important:
- On Red Hat® OpenShift® Container Platform 4.21 or later, the Global Data Platform service is not available as a separate service. Global Data Platform logs are collected as part of Fusion Data Foundation.
- Enable scale traces before you collect the Global Data Platform logs. For more information about how to enable traces, see Generating GPFS trace reports.

Use this option to collect logs related to the physical storage configuration of IBM Fusion HCI and Global Data Platform. These logs are used to troubleshoot storage issues. For example, issues like loss of storage access, problems with CSI provisioning, problems related to disaster recovery (Metro-DR and Regional DR), problems with the overall health of the Global Data Platform.

Backup & Restore
Use this option to collect logs related to the Backup & Restore service. These logs are used to troubleshoot issues that are related to setup, backup, and restore. These logs contain mainly the `ibm-backup-restore` namespace resources.
For Backup & Restore service, you can also select logs for the specific clusters as follows:

a. In the Summary slide out pane, click edit icon next to Backup & Restore.
The Select spoke clusters page gets displayed with all the available clusters.

b. Select the spoke clusters that you want to include in the log package.
Important:
- You can also collect logs for spokes from Hub cluster by selecting the spoke clusters from the Hub cluster IBM Fusion HCI user interface.
- These logs can also be collected from the individual Spoke clusters.

IBM Data Cataloging
Use this option to collect logs that are related to the IBM Data Cataloging service. These logs contain mainly the `ibm-data-cataloging` namespace resources.

Data Foundation
Use this option to collect logs that are related logs related to Kubernetes resources of openshift-local-storage, openshift-storage-client, openshift-lvm-storage, and openshift-storage.

Content-Aware Storage
Use this option to collect logs related to the Content-Aware Storage service. These logs contain mainly the `scale must-gather` and `ibm-cas-namespace` resources.

Important: If a service is not enabled, the user interface does not display the service component in the log collection.

4. Click Create.
The created log set gets added to the log list with status as Creating. After the collection is complete, the status changes to Success.
The Support logs page lists all created logs with the following details:

Name
: The name of the collection set.
Note: The name of the log collection set consists of two parts: component name and time stamp. If you collect log collection sets for the storage, for example, the file name must be sc-20220324124442. The name of the log collection set change for every component.
- `Storage: sc`
- `Data foundation: odf`

Contents
: It contains details of which component logs comprise the collection.

Status
: The values of the status field are Collecting, Success, and Partial.

Created
: The date and time of the creation of the log set.

Expires
: The expiry date and time of the log set.

Note: The collected logs are auto-deleted after 24 hours.

5. After a log package is successfully added to the log list, you can choose to upload the logs to IBM directly by using the Call Home feature or download from the browser and review the logs. If you have not enabled Call Home, then Upload log package to IBM via Call Home option is disabled.
Note: The Partial status indicates that the log collection completed with errors. You can upload and download Partial logs.
For steps to enable call home, see Enabling Call Home.
   a. Click the ellipsis menu of the log record and click Upload.
   The Upload log package window gets displayed. The collected logs get uploaded so that the IBM Support team can use for diagnosis.
   b. In the Upload log package window, choose whether you want to create a new support case or attach to an existing case.
   c. If it is an existing case, then enter the Ticket number.
   Note: A log collection package can be uploaded only once to any ticket through this user interface page.
   d. Click Upload log.

Download collected log package
: Click the ellipsis menu of the log record and click Download. The Downloading log package window gets displayed indicating that the download is in progress.

Delete a log package
: a. Click the ellipsis menu of the log record and click Delete.
b. In the Confirm delete log package window, click Delete.

If you have not enabled Call Home, then download the log package and upload it to an IBM site. In the table of log packages, click the ellipsis menu for the log package that you wish to upload and select the Download action. It triggers the download of the log package file in your web browser. The user interface provides three methods of uploading logs to IBM.

FTP transfer
: This option is the fastest approach to upload logs to IBM. It uses FTP to upload logs to the IBM Enhanced Customer Data Repository (ECUREP) site.

Browser upload
: With the browser upload option, you can upload the log package through your web browser. This option is limited to files that are smaller than 200 MiB.

Blue Diamond
: HIPPA customers designated as Blue Diamond need to upload logs to the IBM Blue Diamond site.

- **Log package**
This section provides a quick overview of the log packages of the IBM Fusion.
- **Automatic cleanup of log packages**
This section describes the automatic cleanup feature for diagnostic log packages that are collected from the IBM Fusion environment.

Log package

This section provides a quick overview of the log packages of the IBM Fusion.

The following log packages are available in the Logs page:

1. Global Data Platform
2. Data Foundation
3. Backup and restore
4. Data Cataloging
5. Content-Aware Storage

Every log package contains a HealthManagerCR folder expect for the Backup & Restore service.

Global Data Platform

Logs related to storage availability. The Global Data Platform log package mainly contains the folders for the following namespaces:
Important: By default, it collects the logs from the IBM Fusion namespace and the namespace where the service is deployed.

- `Fusion namespace`
- `ibm-spectrum-scale-namespace`
- `ibm-spectrum-scale-csi-namespace`
- `ibm-spectrum-scale-operator-namespace`

Global Data Platform log packages contain both the custom and cluster resources. The custom resources such as **spectrumfusions**, **scalemanagers**, **scaleclusters**, and **nodes**. The cluster resources such as **storageclasses.**

Follow the steps to collect **Velero** namespace logs:

1. On the Global Data Platform service, manually add the **Velero** namespace under the namespaces in the serviceability section of the **FusionServiceDefinition** instance.
2. It ensures that the **Velero** namespace logs are collected as part of log collection for the Global Data Platform service.
3. The **Velero** namespace is specified in **ramen-dr-cluster-operator-config** configmap in the **ibm-spectrum-fusion-ns** namespace.

It also includes the scale must-gather folder with the relevant log files.

Data Foundation

Logs related to storage availability. The Red Hat® OpenShift® Data Foundation log package mainly contains the folders for the following namespaces:
Important: By default, it collects the logs from the IBM Fusion namespace and the namespace where the service is deployed.
It includes the **Openshift-storage** and **Openshift-local-storage** folder with the relevant log files.
Note: Data Foundation log status shows partial if any of the namespaces are not available, but you can upload and download partial logs.
Red Hat OpenShift Data Foundation log packages contains both the custom and cluster resources. The custom resources such as **spectrumfusions**, **odfmanagers**, **odfclusters**, **fusionservicedefinitions**, **fusionserviceinstances**, **storageclusters**, **cephclusters**, **localvolumediscoveries**, **localvolumediscoveryresults**, **localvolumesets**, **catalogsources**, **volumesnapshotclasses**, and **clustercsidrivers**.

The cluster resources such as **persistentvolumes**, **clusterrolebindings**, **storageclasses**, and **storageclassclaims**.

Follow the steps to collect **Velero** namespace logs:

1. On the Data Foundation service, manually add the **Velero** namespace under the namespaces in the serviceability section of the **FusionServiceDefinition** instance.
2. It ensures that the **Velero** namespace logs are collected as part of log collection for the Data Foundation service.
3. The **Velero** namespace is specified in **ramen-dr-cluster-operator-config** configmap in the **ibm-spectrum-fusion-ns** namespace.

Backup and restore

Logs related to backup and restore of your workload applications. The backup and restore log package mainly contains the folders for the following namespace.
Important: By default, it collects the logs from the IBM Fusion namespace and the namespace where the service is deployed.

ibm-backup-restore
: It contains the logs and object definition files for all of the major resources under this namespace. For example, routes, pods logs, deployments, daemon sets, persistent volumes, and config maps.
Note: It includes the Kubernetes events that are associated with the resource. If any resource does not exist, then it can be skipped.

If Cloud Pak for Data (CPD) is installed and custom resource (CR) is created, the backup and restore log packages also include backup and restore–specific CPD logs, which contain log data from all CPD-related namespaces.
Note: The CPD log collection process might fail to collect logs for Velero backup and restore operations. The failure occurs when the CPD script is unable to validate or recognize self-signed TLS certificates configured for the backup storage location. To prevent this, ensure that a properly configured and trusted TLS certificate is associated with the backup storage location.

Data Cataloging

Logs related to Data Cataloging service. The Data cataloging log package mainly contains the folders for the following namespace.
Important: By default, it collects the logs from the IBM Fusion namespace and the namespace where the service is deployed.

ibm-data-cataloging
: It contains the logs and object definition files for all of the major resources under this namespace. For example, routes, pods logs, deployments, daemon sets, persistent volumes, and config maps.
Note: It includes the Kubernetes events that are associated with the resource. If any resource does not exist, then it can be skipped.

Content-Aware Storage

Logs related to Content-Aware Storage service. The Content-Aware Storage log package mainly contains the **scale must-gather** and **ibm-cas-namespace** resources.
Important: By default, it collects the logs from the IBM Fusion namespace and the namespace where the service is deployed.

Automatic cleanup of log packages

This section describes the automatic cleanup feature for diagnostic log packages that are collected from the IBM Fusion environment.

After a log package collection completes, the system automatically removes or sanitizes sensitive information before the package is available to download. This behavior protects sensitive data and supports compliance with enterprise security and data-protection requirements.

Key benefits

- Protects sensitive information by removing or masking confidential data.
- Supports compliance with data-protection and privacy requirements.
- Enables safe sharing of log packages with IBM Support.
- Operates automatically without requiring additional user actions.
- Preserves diagnostic information that is needed for analysis and troubleshooting.

Protected information

The log cleanup process removes or obfuscates sensitive data in collected log packages.

Removed data

- Kubernetes secrets, including `secret` resources and `secret.yaml` files
- Credentials such as passwords, tokens, and API keys
- Authentication files, including `kubeconfig`, pull secrets, and service account tokens
- Private keys, including SSH keys and TLS private keys (`.key`, `.pem`)
- Certificate files, including SSL and TLS certificates (`.crt`)

Masked or obfuscated data

- IP addresses are replaced with consistent placeholder values
- MAC addresses are replaced with consistent placeholder values
- Internal domain names are obfuscated
- Credentials (API keys) and tokens are obfuscated
- PEM certificate content found in files

Unchanged content

The cleanup process removes or obfuscates only sensitive information. All logs, resources, and configuration data that is required for debugging remain intact.

User experience

By default, automatic log cleanup is enabled and requires no manual intervention.

1. Collect logs by using the IBM Fusion UI.
2. The system automatically sanitizes sensitive data in the log package.
3. Download the clean and secure log package.

Configuration

When to change the configuration (optional)

The default configuration is secure and recommended for all environments. Disable automatic log cleanup only in the following scenarios:

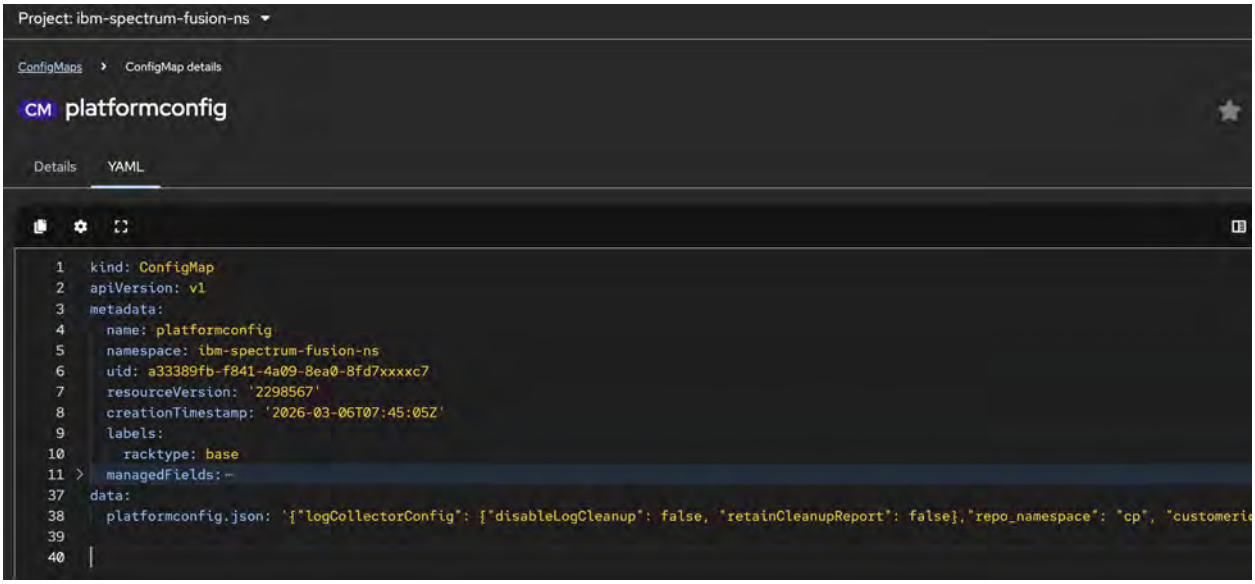
- Logs remain strictly within the organization for internal debugging
- IBM Support explicitly requests for unsanitized log packages

Disabling automatic log package cleanup

To disable automatic log package sanitization by using the OpenShift console:

1. Log in to the OpenShift® web console.
2. Switch to Project `<FusionNamespace>` (typically `ibm-spectrum-fusion-ns`).
3. Go to Workloads > ConfigMaps, search for `platformconfig` ConfigMap, and open it in YAML view.
4. Locate the `logCollectorConfig` entry in the ConfigMap and update it with the following configuration. If the `logCollectorConfig` entry does not exist, add the following entry to the JSON content of the key `platformconfig.json`:

   ```
   "logCollectorConfig": {"disableLogCleanup": true},
   ```



5. Save the changes.
   The automatic log package cleanup is now disabled.

Benefits of enabling Call Home

Configure and enable the call home feature as it improves the user experience when IBM Fusion is used.

The call home feature when enabled provides the following benefits:

1. Improves customer service response time
   - *Scheduled data uploads:*
     If the Call Home feature is enabled, then the IBM® support representatives can start with the analysis of the issue that is reported without any delay. The IBM support team can use the cluster data from the Call Home scheduled uploads to do an immediate analysis of the issue. For more information, see Collecting logs in IBM Fusion.

     However, if the Call Home feature is not enabled, then the IBM support team first collects debugging data from the IBM Fusion user interface. The whole process takes time to request, collect, and deliver the data, which might cause a delay in resolving the issue on the site.

   - *Selected failure events trigger automatic Call Home uploads:*
     If the Call Home feature is enabled, then the event-based uploads feature automatically collects the relevant component-specific information and uploads it to the IBM ECuRep server. The uploaded data enables the IBM support team to immediately find the root cause of the issue that is reported on the customer site.

2. Proactively detects issues
   - Detects violations against best practices and common misconfiguration.
   - Finds out which specific customer site is affected by the reported issue by using tools like High Impact Programming Error Authorized Program Analysis Report (HIPER APARs).
   - Constantly extends the list of automatically detected issues, best practice violations, and common misconfiguration over time.

   Note: IBM Fusion user interface displays the list of events.
3. Improves customer support experience in areas of consumability and ease of usage
   - *Uploads gpfs snaps directly from the IBM Fusion system:*
     If the Call Home feature is enabled, then a customer can find out the customer number. The requested debugging data is transferred from the customer site cluster or node to the IBM support team by raising a service ticket. For more information, see Enabling Call Home.

4. Creates service tickets automatically for reported failures
   If the call home feature is enabled, then the system automatically creates Salesforce service tickets for any hardware failures that are reported on customer sites.
   Note: A user can configure the Call Home settings on the IBM Fusion user interface.

- **Data privacy with Call Home**
  A licensee can configure the IBM Fusion to automatically send specific cluster information to the IBM support team by using the Call Home feature.
- **Enabling Call Home**
  You can enable Call Home from the Support settings page, which automatically uploads health telemetry and error logs to IBM Support. Call Home can be enabled only when the contact information is provided.
- **Configuring Call Home with proxy**
  You can configure Call Home to use a proxy server for outbound connectivity to IBM Support.
- **IBM Fusion Call Home Telemetry**
  IBM Fusion Call Home Telemetry enhances supportability by providing IBM with valuable insights into system configuration and performance. Telemetry data is collected from various components and securely transmitted to IBM, helping support teams proactively identify issues and optimize system behavior based on real-world usage.
- **Custom Certificate Authority (CA) support for Call Home**
  Support for custom Certificate Authority (CA) for IBM Fusion Call Home.

Data privacy with Call Home

A licensee can configure the IBM Fusion to automatically send specific cluster information to the IBM® support team by using the Call Home feature.

When a licensee enables the Call Home feature, the licensee provides the support contact information, such as names, phone numbers, or email addresses, which are needed during the Call Home feature configuration and activation process.

By default, when a licensee decides to activate and use the Call Home feature, the licensee agrees to allow IBM and its subsidiaries to store and use the licensee's support contact information. This information is processed and used with IBM business relationship and might be shared with third-parties under the direction of IBM. For example, IBM support center representatives or assignees of IBM and its subsidiaries.

The support contact information can be used for processing business orders, business promotions, problem determination, or market research.

When the call home feature is configured and activated, the IBM Fusion collects all monitoring information that is related to system utilization, performance, capacity planning, and service maintenance. The service information includes system failure logs, part numbers, machine serial number, software version, maintenance levels, installed patches, and configuration values.

When the Call Home feature is enabled, the licensee allows IBM to use and share the data, which is gathered from system monitoring functions, within IBM and with IBM business partners and third-parties, such as IBM support centre sub-contractors or assignees. The shared data is used only under the direction of IBM for the following defined purposes:

- Determining a problem.
- Assisting licensee with performance and capacity planning.
- Assisting IBM to enhance IBM products and services.
- Notifying licensee about the licensee's system status and available solutions.

Note: The licensee information excludes the collection and transmission of licensee's financial, statistical, and personal data, and licensee's business plans.
When the Call Home feature is enabled, a licensee agrees to the fact that the licensee's support contact information can be transferred to countries that might not be a member of the European Union. A licensee can disable the Call Home feature at any time.

Enabling Call Home

You can enable Call Home from the Support settings page, which automatically uploads health telemetry and error logs to IBM Support. Call Home can be enabled only when the contact information is provided.

Before you begin

Ensure that the firewall sites are configured for Call Home:

- `.esupport.ibm.com`
- `.ecurep.ibm.com`

Configure your data center's firewall rules or proxy access control list (ACL) to allow the manual and automated Service and Support request feature to communicate with the servers. To configure firewall, consider the following points for planning and prerequisites:

- Source IP for Call Home outbound connections is the OpenShift® control node.
- The Host name, IP, and port details of Call Home servers:

  Table 1. IP addresses of the IBM call home servers

Host name	IP address	Ports	Protocol
esupport.ibm.com	192.148.6.11	443	https
ecurep.ibm.com	192.109.81.21	443	https

  Important: Though the table gives IP address guidance, ensure that you check the following pages for latest information: IP addresses and https://www.ibm.com/support/pages/node/6853429.

If your environment requires a proxy, confirm the proxy details and credentials before proceeding.

Note: Telemetry is no longer configured as a separate option. When Call Home is enabled, telemetry collection and transmission are enabled automatically. For more information, see IBM Fusion Call Home Telemetry.

About this task

Support status banner

A yellow banner might appear at the header of the Support settings page with messages similar to:

- `Support features disabled. Call Home or other support features are not enabled. Review support settings to restore full support capabilities.`
  This banner is an expected behavior in the following case:
  - Call Home and/or telemetry is disabled
- `Support disconnected. One or more support features are not functioning. Review your support settings to restore full support capabilities.`
  This banner is an expected behavior in the following cases:
  - Call Home or telemetry is enabled but in a failed or degraded state
  - Required contact information is missing or incomplete

After Call Home is successfully enabled and connectivity is restored, the banner is cleared automatically.

Procedure

1. Open Support settings.
   In the IBM Fusion user interface, go to Support >Support settings.

   Note: If contact information is not configured, the page displays a warning: `Contact details must be provided before enabling any support settings.` The Enable button for Call Home is disabled until this information is added.
2. Add contact information.
   a. In the Contact information section, click Add.
   b. Enter contact information:
      In the Contact information dialog, provide the primary support contact details:
      i. Enter Company name.
      ii. Enter Customer number.
      iii. Enter Primary contact name.
      iv. Enter Primary email address.
      v. Enter Primary phone number:
         1. Select the country code.
         2. Enter the phone number.
   c. Click Add to save the information.
      After saving, the contact information is displayed on the Support settings page, and the Call Home option becomes available.
3. Enable Call Home.
   a. In the Call Home section, click Enable.
   b. Select a connection type:
      - Direct connection, or
      - Proxy connection, if required by your network.
   c. If you are using a proxy, enter the proxy configuration details:
      - Host and port
      - Authentication details or certificates, if required.
   d. Click Test connection to validate connectivity.

Results

- If the test succeeds, a success confirmation is displayed and Call Home is enabled.
- If the test fails, an error message explains the issue (for example, proxy authentication failure or connectivity error). Correct the configuration and test again.

What to do next

After successful configuration:

- The Call Home section shows an enabled status.
- Any connectivity or configuration issues are surfaced directly on the Support settings page.

These steps ensure that Call Home is correctly enabled and ready to support proactive monitoring and faster issue resolution.

Configuring Call Home with proxy

You can configure Call Home to use a proxy server for outbound connectivity to IBM Support.

About this task

Call Home supports both HTTP and HTTPS proxies, including HTTP, HTTPS (TLS), and HTTPS with mutual TLS (mTLS). Select the connection type based on your environment's ability to allow outbound internet access.

Call Home supports direct connectivity and different proxy configurations to align with common enterprise network and security models.

Important: The Proxy host field must include the protocol: `http://` or `https://`. A hostname without the protocol prefix is not accepted. For example: `http://proxy.example.com` or `https://proxy.example.com`.

Procedure

1. In the IBM Fusion user interface, go to Settings > Support.
2. Review and complete the contact information.
3. In the Call Home section, click Edit.
4. Select a connection type.
   Choose the appropriate connection type based on your network configuration:

   Direct connection
   : Use a direct connection when the system can reach IBM Support endpoints directly without routing through a proxy. No additional fields are required for this option.

   Open proxy
   : Use an open proxy when outbound traffic must pass through a proxy that does not require authentication. This option supports both HTTP and HTTPS proxies. Provide a Certificate Authority (CA) certificate only if the proxy uses a non-public CA.

   Basic authentication
   : Use basic authentication when the proxy requires a username and password for access. This option supports HTTP and HTTPS proxies. If HTTPS is used and a private CA issues the proxy certificate, a CA certificate must be provided.

   Certificate authentication
   : Use Certificate authentication when the proxy uses certificate-based security. This option supports TLS and mutual TLS (mTLS), allowing configuration of the CA certificate for server validation and, when required, a client certificate and private key. It is typically used in environments with stricter security or compliance requirements. If required by the proxy, username and password can also be provided.

5. Configure the proxy details as required.
   The following fields apply whenever a proxy is configured, regardless of authentication type:

Field	Required	Description
Proxy host	Yes (for proxy connections)	Fully qualified proxy URL. It must start with `http://` or `https://`.
Port	Yes (for proxy connections)	Network port on which the proxy listens (for example, 8080 or 3128).
Username	Optional	Proxy username, if required by the proxy configuration.
Password	Optional	Proxy password, if required by the proxy configuration.
CA certificate	Conditional	Certificate Authority (CA) is used to validate the proxy server certificate. Required when the proxy uses HTTPS (TLS or mTLS) with a private or internal CA. Supported formats: `.crt`, `.cer`, `.cert`, `.ca`
Client certificate	Conditional	A client certificate is presented to the proxy during mutual TLS (mTLS). Required only when the proxy enforces client certificate authentication. Supported formats: `.crt`, `.cer`, `.cert`, `.ca`
Client key	Conditional	Private key corresponding to the client certificate. Required only for mTLS authentication. Supported formats: `.pem`, `.key`, `.crt`, `.cer`, `.cert`, `.ca`

6. Save and test the configuration.
   a. Click Save to apply the configuration.
   b. Click Test connection to validate the configuration.
   - If the test succeeds, a success confirmation is displayed and Call Home is configured with the proxy settings.
   - If the test fails, an error message explains the issue. Verify the proxy configuration and test again.

IBM Fusion Call Home Telemetry

IBM Fusion Call Home Telemetry enhances supportability by providing IBM with valuable insights into system configuration and performance. Telemetry data is collected from various components and securely transmitted to IBM, helping support teams proactively identify issues and optimize system behavior based on real-world usage.

Telemetry automatically collects and transmits the data from IBM Fusion components, such as software applications or hardware devices. Also, it is used to gather information about system configuration, usage patterns, and performance metrics. For steps to enable call home telemetry, see Enabling Call Home.
Note: The system uploads Call Home telemetry data every 24 to 26 hours.
The following details are collected and uploaded as part of call home telemetry support:

IBM Fusion cluster details
: Collects statistical insights into IBM Fusion resources, such as the number of pods, deployments, and other key components, along with version and health status details for both OpenShift® and IBM Fusion. Also, it collects node storage consumption details for OpenShift.

IBM Fusion license data
: Collects IBM Fusion licensing data. For more information on licensing metrics, see Understanding the IBM Fusion license report.

IBM Fusion events data
: Collects the summary of IBM Fusion events by severity and event ID.

IBM Fusion Services
: Collects the details such as name, version, health, and upgrade details of different fusion services installed in the cluster. If Backup & Restore service is installed, then it shows the success % of backups and restores.

Hosted clusters
: Collects basic count and health status of hosted clusters that are deployed.

Virtual machines
: Collects basic count and health status of virtual machines that are created on cluster.

Custom Certificate Authority (CA) support for Call Home

Support for custom Certificate Authority (CA) for IBM Fusion Call Home.

Overview

IBM Fusion Call Home supports custom Certificate Authorities (CAs) to enable seamless operation in enterprise environments that use TLS-intercepting proxies. This support allows Call Home to establish secure connections to IBM support services even when your network uses non-standard certificate authorities.

Importance of custom CA support for Call Home

In many enterprise environments, standard TLS certificates do not meet security policy or network architecture requirements. Custom CA support addresses these challenges by enabling Call Home to trust certificates signed by your organization's certificate authorities.

Use cases and scenarios

Custom CA support is required in the following scenarios:

TLS-intercepting proxies
: Many organizations use security proxies that intercept outbound HTTPS connections to inspect traffic for threats. These proxies decrypt the original connection, inspect the content, and then re-encrypt it using an enterprise certificate before forwarding to the destination.
Example flow:

```
Call Home  →  Corporate proxy  →  IBM support
              (intercepts and     (esupport.ibm.com)
               re-signs with
               enterprise CA)
```

Without custom CA support, Call Home rejects the re-signed certificates as untrusted, which prevents communication with IBM support services.

Private certificate authorities
: Organizations that use private Certificate Authorities sign certificates for internal services and infrastructure. When Call Home communicates through infrastructure secured by these private CAs, it must trust the certificate chain to establish secure connections.

Custom CA integration behavior

Call Home integrates with OpenShift® cluster-wide CA configuration through an automated process:

- Automatic discovery: Call Home automatically detects and uses custom CAs configured at the OpenShift cluster level.
- Dynamic updates: When cluster administrators update CA configurations, Call Home automatically receives and applies these changes.
- Zero configuration: Call Home does not require manual CA configuration.
- Operator-managed: The ISF serviceability operator manages CA injection and updates.

Configuring License Service Reporter for IBM Fusion

Configure License Service Reporter to obtain a unified view of IBM Fusion licensing metrics. License Service Reporter supports this feature and can be extended to IBM Fusion. In addition to the IBM Fusion user interface, it provides an additional view of licensing metrics, offering deeper insights. It can also be configured across multiple clusters, enabling centralized monitoring and reporting of license usage in distributed environments.

Before you begin

- Ensure that the newest version of License Service Reporter is installed, with support for Usage Metering Operator as a data source.
- Review the prerequisites for IBM License Service Reporter in the Installing and configuring License Service Reporter section.

Install and configure License Service Reporter

Ensure that the IBM License Service Reporter is installed on the cluster. If it is not installed, follow the instructions in the Installing License Service Reporter from public container registry section.

Configure a connection to the License Service Reporter

Follow the steps in the Configuring a connection to License Service Reporter by using the OpenShift console section to connect the Usage Metering Operator in the Fusion namespace to License Service Reporter.

Use the following high-level steps and documentation links to simplify the setup process for IBM Fusion.

Create the following resources in the namespace where IBM Fusion is installed:

Create the ConfigMap

Edit the following YAML to match your environment. Use the URL and certificate that is provided by License Service Reporter.

```
kind: ConfigMap
apiVersion: v1
metadata:
 name: <name>
data:
  url: '<License Service Reporter URL>'
  tls.crt: |
    <License Service Reporter certificate>
```

For more information on obtaining the License Service Reporter's URL and certificate, see Creating the ConfigMap.

Create the secret

To create the secret that is used to configure the connection to the License Service Reporter, update the following YAML to be in line with your environment:

```
kind: Secret
apiVersion: v1
metadata:
  name: <name>
data:
  token: <License Service Reporter token>
type: Opaque
```

For more information on retrieving the License Service Reporter token, see Creating the secret.

Create the configuration Custom Resource (CR)

1. Log in to the OpenShift® console of the cluster where IBM Fusion is deployed.
2. Open the IBM Fusion namespace.
3. Go to Operators > Installed Operators.
4. Select IBM Usage Metering Operator and open the IBM Usage Metering tab.
5. Click ibm-usage-metering-fusion and open the YAML tab.
6. Add the following content to the `spec` section and save:

```
spec:
   sender:
     reporter:
       configMapName: <Name of the ConfigMap>
       secretName: <Name of the secret>
```

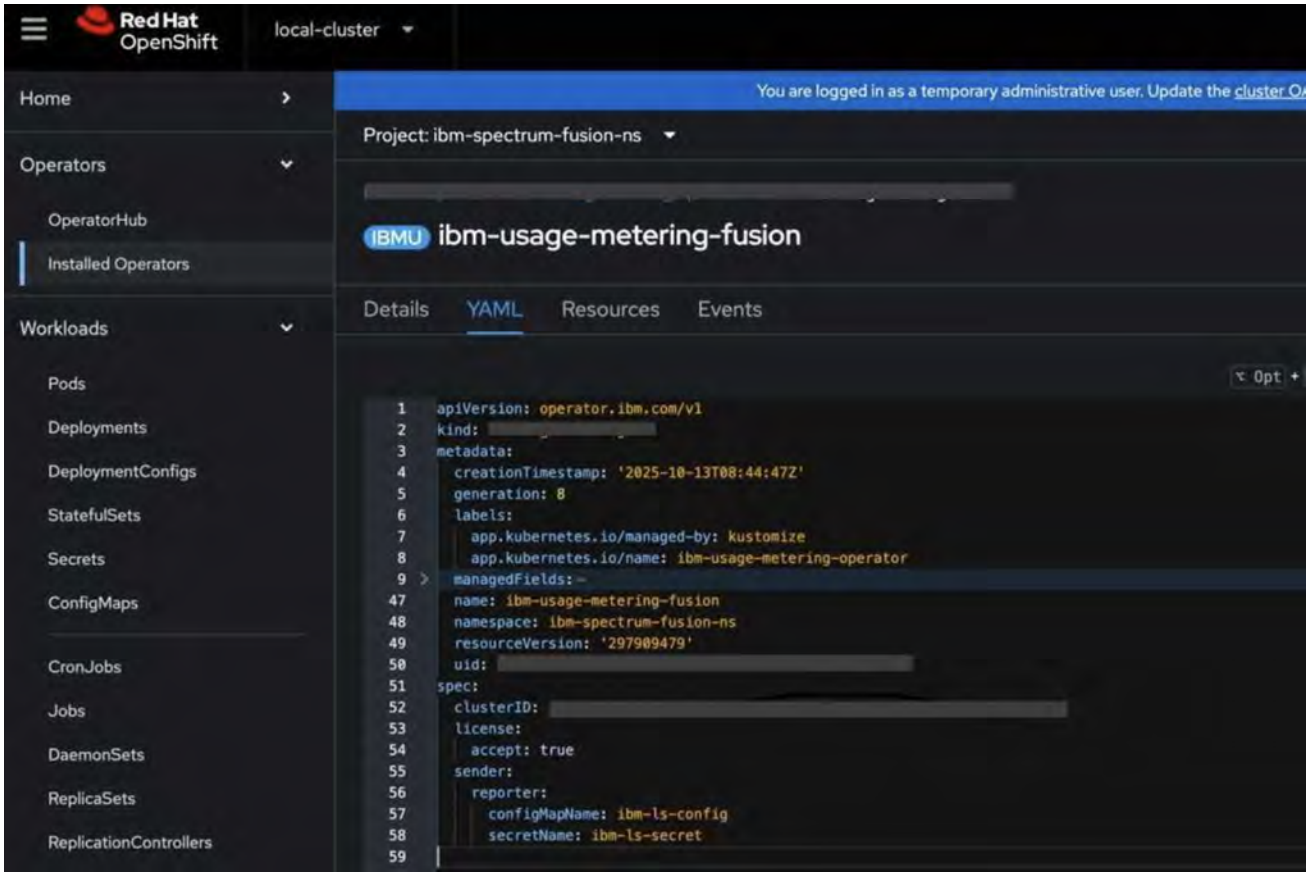
Note: The `configMapName` and `secretName` are the names of the ConfigMap and secret that are created in the previous steps.

Results

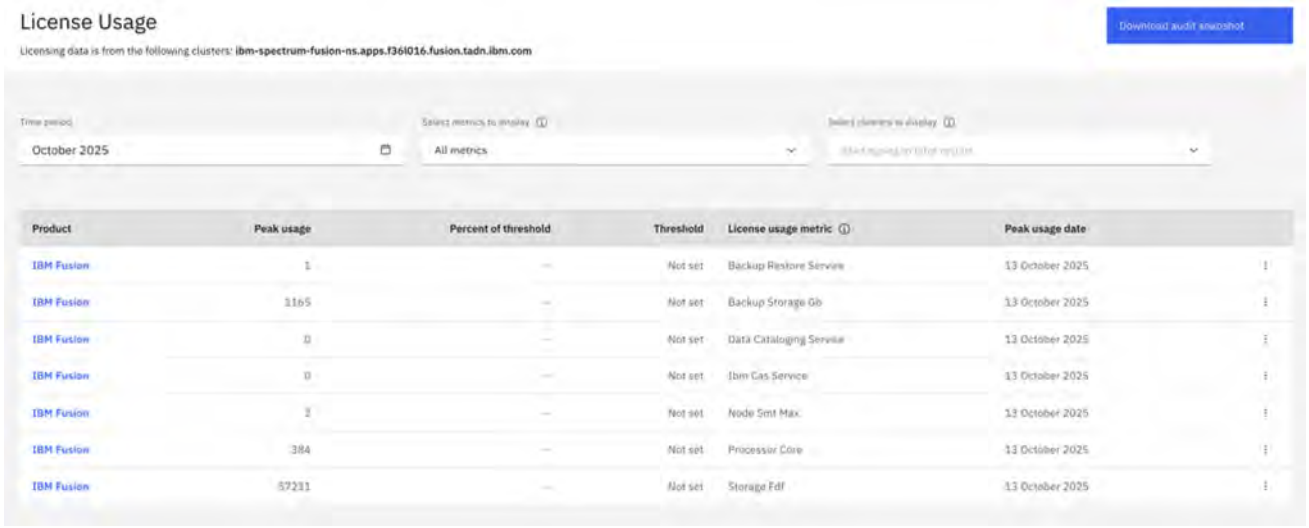
After completing the configuration, a connection is set and the Usage Metering Operator begins sending data to License Service Reporter. It enables a unified view of IBM Fusion licensing metrics.

Example of the unified view of IBM Fusion Licensing metrics in License Service Reporter.

- **Usage metering operator setup**



- **License Service Reporter setup**



What to do next

To log in to License Service Reporter console, follow these steps:

- For more information on getting the console route, see Retrieving License Service Reporter console route.
- For more information on how to retrieve the credentials to login, see Retrieving credentials for basic authentication.

Events and error codes message references in IBM Fusion

This reference information provides a consolidated view of all the events or error messages that can be displayed when the IBM Fusion system might results in an event or an error.

About event or error messages

When you encounter an event or error message in a log or on other parts of the IBM Fusion user interface, look up the event or error message by its prefix to find more information. Use the events and error messages to diagnose and solve problems when you troubleshoot the IBM Fusion system.

Message prefix format

A message number format denotes the type of message and with which component it is associated with. The default prefix starts as `BMY` in the code, which is followed by one of these tags:

- The installation-related events tag is associated with `IN` in the message code.
- The backup and restore-related events tag is associated with `BR` in the message code.
- The call home related events tag is associated with `CH` in the message code.
- The storage related events tag is associated with `ST` in the message code.
- The 4 digits after the tag denote the number of the event or error message. For example, `0000`.
- The storage events tag is associated to `BMYSTxxxx` in the message code.
- The list of events that can open tickets tag is associated to `BMYXC` in the message code.

An example of a message prefix format is `BMYST0001`.

Message severity tag

A severity tag is a one-character alphabetic code, followed by a message. You can determine the message severity by examining the text or by contacting the IBM® Support. To collect logs, see Collecting logs in IBM Fusion.

Table 1. Message severity tags

Severity tag	Message type	Meaning
C	CRITICAL	Indicates a critical condition that must be corrected immediately. The system discovered an internal inconsistency of some kind. The system functioning might be halted or the system might attempt to continue despite the inconsistency. Report these errors to IBM® Support.
I	INFORMATIONAL	Indicates normal operation. This message by itself indicates that nothing is wrong; no action is required.
W	WARNING	Indicates a problem, but the system functioning continues. The problem can be a transient inconsistency. It can be that the action skipped some operations on some objects, or is reporting an irregularity that could be of interest.

- **Install events and error codes**
  List of all the events that you might encounter during installation.
- **Upgrade events and error codes**
  List of all the events that you might encounter during upgrade.
- **Global Data Platform events and error codes**
  List of all the events and error codes that you might encounter while you work with storage.
- **Data foundation events and error codes**
  List of all the events and error codes that you might encounter while you work with Data Foundation storage.
- **IBM Data Cataloging events and error codes**
  List of all the events that you might encounter while you work with IBM Data Cataloging for the IBM Fusion.
- **Backup & Restore events and error codes**
  List of all the events that you might encounter during Backup & Restore for the IBM Fusion.
- **Connection services events and error codes**
  List of all the events that you might encounter during connection service for the IBM Fusion.
- **Content-Aware Storage events and error codes**
  List of all the events that you might encounter while you work with CAS for the IBM Fusion.
- **Serviceability events and error codes**
  List of all the events that you might encounter that related to serviceability.

Install events and error codes

List of all the events that you might encounter during installation.

For Informational, Warning and Critical events, see the following:

- **BMYIN1501**
  Application has been created successfully.
- **BMYIN1502**
  Installing service Global Data Platform.
- **BMYIN1503**
  Installed service Global Data Platform.
- **BMYIN1504**
  Installing service Backup & Restore.
- **BMYIN1505**
  Installed service Backup & Restore.
- **BMYIN1506**
  Installing service Data Foundation.
- **BMYIN1507**
  Installed service Data Foundation.
- **BMYIN1508**
  Service is installed.
- **BMYIN1509**
  Service installation is complete.

- **BMYIN1510**
  Service instance is discovered.
- **BMYIN1512**
  Catalog source is created.
- **BMYIN1513**
  Catalog source creation is completed.
- **BMYIN1514**
  Discovered service Data Foundation.
- **BMYIN2501**
  Error installing service Global Data Platform.
- **BMYIN2502**
  Error installing service Backup & Restore.
- **BMYIN2503**
  Error installing service Data Foundation.
- **BMYIN3116**
  Service is Unhealthy.
- **BMYIN3117**
  Service Catalog source creation is failed.
- **BMYIN3119**
  Service installation failed.
- **BMYIN4101**
  `UnsupportedPlatform`
- **BMYIN4102**
  `RelatedServiceInstalled`
- **BMYIN4103**
  `RelatedStorageServiceInstalled`
- **BMYIN4104**
  `StorageClassPrereqNotFound`
- **BMYIN4105**
  `OCPVersionPrereqNotFound`
- **BMYIN4106**
  `CRDMissingPrereqNotFound`
- **BMYIN4107**
  `DataJSONKeyNotFoundInCM`
- **BMYIN4108**
  `SkipServiceInSpectrumFusionCRTrue`
- **BMYIN4110**
  `PrereqServiceNotInstalled`
- **BMYPC5010**
  IBM Fusion operator installation is in progress.
- **BMYPC5027**
  The OpenShift® Container Platform is currently operating at the latest version that is supported by IBM Fusion.
- **BMYPC6006**
  A pod in the `openshift-operator-lifecycle-manager` namespace is not ready.
- **BMYPC6007**
  A `Catalogsource` is not ready.
- **BMYPC6008**
  Inaccessible registry.
- **BMYPC6010**
  IBM Fusion operator is in a `Failed` state or not found.
- **BMYPC6013**
  The DNS resolution for image registry failed.
- **BMYPC6040**
  Backup & Restore agent instance not found.
- **BMYPC6042**
  The Subscription is missing the `CatalogSource`.
- **BMYPC6044**
  Backup & Restore server instance is unhealthy.
- **BMYPC6045**
  Backup & Restore agent instance is unhealthy.

BMYIN1501

Application has been created successfully.

Severity

Informational

User actions

No action is required.

BMYIN1502

Installing service Global Data Platform.

Severity

Informational

User actions

No action is required.

BMYIN1503

Installed service Global Data Platform.

Severity

Informational

User actions

No action is required.

BMYIN1504

Installing service Backup & Restore.

Severity

Informational

User actions

No action is required.

BMYIN1505

Installed service Backup & Restore.

Severity

Informational

User actions

No action is required.

BMYIN1506

Installing service Data Foundation.

Severity

Informational

User actions

No action is required.

BMYIN1507

Installed service Data Foundation.

Severity

Informational

User actions

No action is required.

BMYIN1508

Service is installed.

Severity

Informational

User actions

No action is required.

BMYIN1509

Service installation is complete.

Severity

Informational

User actions

No action is required.

BMYIN1510

Service instance is discovered.

Severity

Informational

User actions

No action is required.

BMYIN1512

Catalog source is created.

Severity

Informational

User actions

No action is required.

BMYIN1513

Catalog source creation is completed.

Severity

Informational

User actions

No action is required.

BMYIN1514

Discovered service Data Foundation.

Severity

Informational

User actions

No action is required.

BMYIN2501

Error installing service Global Data Platform.

Severity

Warning

User actions

Collect Storage logs and contact IBM support.

BMYIN2502

Error installing service Backup & Restore.

Severity

Warning

User actions

Collect Backup and restore logs and contact IBM support.

BMYIN2503

Error installing service Data Foundation.

Severity

Warning

User actions

Collect Data Foundation logs and contact IBM support.

BMYIN3116

Service is Unhealthy.

Severity

Critical

User actions

Do the following steps to diagnose the error:

1. Go to OpenShift® Container Platform web console and check the FusionServiceInstance for the specific service and check the conditions section to find the error.
2. If the issue persists, run the following oc command, and check the status section and debug the error.

   ```
   oc get fusionserviceinstance <instance-name-of-service> -o yaml
   ```

3. If this occurs during installation or post service installation, collect logs for the corresponding service that shows as unhealthy and contact IBM support. Follow the steps to collect logs from IBM Fusion.
   - Go to Services page in IBM Fusion user interface.
   - Click the ellipses menu for the corresponding service.
   - Select the Collect logs button. It downloads all the necessary logs.

BMYIN3117

Service Catalog source creation is failed.

Severity

Critical

User actions

Do the following steps to diagnose and report the problem:

1. Run the following command and try to debug the results.

   ```
   oc describe catalogsource <catlogsource-name> -n openshift-marketplace
   ```

2. If the result says Username/Password missing, manually update the secret for the required registry.
3. If the issue persists, contact the IBM support.

BMYIN3119

Service installation failed.

Severity

Critical

User actions

Do the following steps to diagnose the error:

1. Go to OpenShift® Container Platform web console and check the FusionServiceInstance for the specific service and check the conditions section to find the error.
2. If the issue persists, run the following oc command, and check the status section and debug the error.

   ```
   oc get fusionserviceinstance <instance-name-of-service> -o yaml
   ```

3. If this occurs during installation or post service installation, collect logs for the corresponding service that shows as unhealthy and contact IBM support. Follow the steps to collect logs from IBM Fusion.
   - Go to Services page in IBM Fusion user interface.
   - Click the ellipses menu for the corresponding service.
   - Select the Collect logs button. It downloads all the necessary logs.

BMYIN4101

`UnsupportedPlatform`

Severity

Informational

Problem description

It indicates that the service is not supported on this platform.

User actions

No action is required.

BMYIN4102

`RelatedServiceInstalled`

Severity

Informational

Problem description

It indicates that the related service is already installed.

User actions

No action is required.

BMYIN4103

`RelatedStorageServiceInstalled`

Severity

Informational

Problem description

It indicates that the related storage service is already installed on this setup.

User actions

No action is required.

BMYIN4104

`StorageClassPrereqNotFound`

Severity

Informational

Problem description

It indicates that the **`StorageClass`** is not present to install this service.

User actions

Check the availability of the storage class specified in the FSD/FSI.

BMYIN4105

`OCPVersionPrereqNotFound`

Severity

Informational

Problem description

It indicates that the OpenShift® Container Platform version is not in line with the service to be installed.

User actions

Upgrade the current OpenShift Container Platform version if services are supported on higher version than the current version.

BMYIN4106

CRDMissingPrereqNotFound

Severity

Informational

Problem description

It indicates that Custom Resource Definition (CRD) is not present on the setup to install the service.

User actions

Check with IBM Support for CRD.

BMYIN4107

DataJSONKeyNotFoundInCM

Severity

Informational

Problem description

It indicates that the ConfigMap information is invalid to install the service.

User actions

Validate whether the ConfigMap is present on the setup and its values are valid, which are specified in the FSD.

BMYIN4108

SkipServiceInSpectrumFusionCRTrue

Severity

Informational

Problem description

It indicates that the service installation is skipped in the **Spectrum Fusion** CR.

User actions

If you want to install the service, you can remove the **skipservice** list values from the **Spectrum Fusion** CR.

BMYIN4110

PrereqServiceNotInstalled

Severity

Informational

Problem description

It indicates that a dependent service is not installed on this setup.

User actions

Install the dependent service first. After the dependent service is installed successfully, the **prereq** service can be deployed.

BMYPC5010

IBM Fusion operator installation is in progress.

Severity

Error

Problem description

The IBM Fusion operator is in **installready** or **installing** or **pending** state, which indicates that the operator install is in progress. It can block the ability of IBM Fusion operator or other IBM Fusion components ability to install, upgrade, or function.

User actions

Do the following steps to resolve the issue:

1. Check for IBM Fusion operator CSV events and conditions to find the root cause for this state. Some of the common reasons for the **Pending** state of the IBM Fusion operator are as follows:
   - Check whether IBM Fusion operator is installing or upgrading.
   - Some changes might corrected in any deployment spec in CSV for IBM Fusion operator. It recreates pods and reconciles to succeeded state in some time.
   - One of the IBM Fusion operator pod is not running. It might be due to an OOM killed error where the pod is requesting more memory than its set limit. To resolve this, update the memory limit for that deployment in CSV.
   - A pod might be in a **crashloopbackoff** due to another reason. Check the pod log and events to find the root cause and take an appropriate action.
   - It is possible that a **Custom Resource Definition** is missing due to an accidental deletion. In such case, contact IBM support to get the CRD YAML and create it in the cluster.

   You can continue with the upgrade operation after the IBM Fusion operator is successful.

BMYPC5027

The OpenShift® Container Platform is currently operating at the latest version that is supported by IBM Fusion.

Severity

Warning

User actions

Do not upgrade OpenShift Container Platform to the next minor version as it is not supported by the IBM Fusion.

BMYPC6006

A pod in the **openshift-operator-lifecycle-manager** namespace is not ready.

Severity

Error

Problem description

It indicates that pod in the **openshift-operator-lifecycle-manager** namespace is not ready. Pods in the **openshift-operator-lifecycle-manager** namespace are responsible for managing catalogs, packages, and operator lifecycle in the OpenShift®. If pods are not running or have errors, then it can cause operator upgrade failure in the IBM Fusion HCI.

User actions

Do the following steps to resolve the issue:

1. Check for the events of the failed pods and logs to find the root cause of failure and take an appropriate action to fix the issue. Pod restart can also resolve the problem.
2. After all the pods in the `openshift-operator-lifecycle-manager` namespace are ready, the upgrade precheck detects the fix and resumes upgrade automatically.

If you want to override the prechecks that are blocking the upgrade, do the following steps:

It converts the `Error` prechecks to `Warning` prechecks and triggers the upgrade.
Note: Before using the `precheck-acks` configmap to override the pre-checks, ensure that the unhealthy resource does not affect the upgrade.

1. Run the following command and export IBM Fusion namespace as an environmental variable.

   ```
   export FUSION_NS="namespace-where-fusion-is-installed"
   ```

2. Run the following command to create a `precheck-acks` configmap in the namespace where the IBM Fusion is installed.

   ```
   oc create configmap precheck-acks -n $FUSION_NS
   ```

3. In the `precheck-acks` config map, add the key `OpenShiftLifeCycleManagementPodStatus` and set the value to `true`.

   ```
   oc patch configmap precheck-acks -n $FUSION_NS --type merge -p '{"data": {"OpenShiftLifeCycleManagementPodStatus": "true"}}'
   ```

BMYPC6007

A `Catalogsource` is not ready.

Severity

Error

Problem description

It indicates that a `Catalogsource` in the OpenShift® is either not ready or not healthy. It is required for any operator lifecycle management operation, such as installation or upgrade to function. All catalog sources in the OpenShift cluster must be healthy and ready. Any unhealthy catalog source impacts the OLM functions in the cluster.

User actions

Do the following steps to resolve the issue:

- Some of the possible reasons with diagnostic steps for a catalog source to not work are as follows:
  1. Invalid image name in the catalog source:
     - Check the pod for the given catalog source in its dedicated namespace or `openshift-marketplace` namespace.
     - Run the following command to describe the pod.

       ```
       oc describe pod <pod name> -n <namespace>
       ```

     - If you find an error with the message `invalid image reference` under events section, then fix the image field in the catalog source YAML. Wait for the catalog source to turn healthy.
  2. Missing images in the registry for disconnected setup:
     - Check the pod for the given catalog source in its dedicated namespace or `openshift-marketplace` namespace.
     - Run the following command to describe the pod.

       ```
       oc describe pod <pod name> -n <namespace>
       ```

     - If you find an error with the message `unknown manifest` under events section, mirror images to enterprise registry and wait for catalog source to turn healthy.
  3. Missing `ImageContentSourcePolicy` for disconnected set up:
     - Check the pod for the given catalog source in its dedicated namespace or `openshift-marketplace` namespace.
     - Run the following command to describe the pod.

       ```
       oc describe pod <pod name> -n <namespace>
       ```

     - If you find an error with the messages `unknown manifest` and `Mirrors failed` under events section, then review the `imagecontentsourcepolicy` applied for any incorrect paths or ports or spelling mistakes. Fix them and wait for catalog source to turn healthy.
  4. Missing credentials for the registry in the global pull-secret of the `openshift-config` namespace. Check the credentials in the global pull-secret in the `openshift-config` namespace:
     - Check the pod for the given catalog source in its dedicated namespace or `openshift-marketplace` namespace.
     - Run the following command to describe the pod.

       ```
       oc describe pod <pod name> -n <namespace>
       ```

     - If you find an error with the messages `Unknown desc = unable to retrieve auth token: invalid` and `username/password: unknown: Authentication is required` under events section, then review the `pull-secret` in `openshift-config` namespace for correct credentials for registry where the image is referenced. Fix auth for registry in secret and wait for catalog source to turn healthy.

5. Connection error to the registry. Ensure that the firewall and proxy do not block registry and port access.
    - Check the pod for the given catalog source in its dedicated namespace or `openshift-marketplace` namespace.
    - Run the following command to describe the pod.

      ```
      oc describe pod <pod name> -n <namespace>
      ```

    - If you find an error with the messages `pinging container registry registry.redhat.io: Get` and `"https://<registryhost>": dial tcp <registryip>: connect: connection refused` under events section, then review the firewall and proxy settings to make sure that registry access is allowed to the cluster and wait for catalog source to turn healthy.

After the `CatalogSource` is in a ready state, the upgrade precheck detects the fix and resumes the upgrade operation automatically.

If you want to override the prechecks that are blocking the upgrade, do the following steps:

It converts the `Error` prechecks to `Warning` prechecks and triggers the upgrade.
Note: Before using the `precheck-acks` configmap to override the pre-checks, ensure that the unhealthy resource does not affect the upgrade.

1. Run the following command and export IBM Fusion namespace as an environmental variable.

   ```
   export FUSION_NS="namespace-where-fusion-is-installed"
   ```

2. Run the following command to create a `precheck-acks` configmap in the namespace where the IBM Fusion is installed.

   ```
   oc create configmap precheck-acks -n $FUSION_NS
   ```

3. In the `precheck-acks` config map, add the key `CatalogSourceStatus` and set the value to `true`.

   ```
   oc patch configmap precheck-acks -n $FUSION_NS --type merge -p '{"data": {"CatalogSourceStatus": "true"}}'
   ```

BMYPC6008

Inaccessible registry.

Severity

Warning

Problem description

It indicates that the specified image registry is not accessible from one or more cluster nodes. It blocks the upgrade because image pulls might fail. It also causes other issues outside of upgrades as newly scheduled or rescheduled pods are unable to access their images.

For the registries that have non-empty auth specified in the pull-secret in the `openshift-config` namespace, check for the connectivity to make sure that the images are pulled successfully for IBM Fusion components and the OpenShift® Container Platform. If you fail to address the registry access might result in pods can get into an `ImagePullbackOff` error.

User actions

Do the following steps to resolve the issue:

1. Check the network connectivity between the OpenShift cluster and the image registry.
2. A firewall can prevent the access to the registry.
3. After the connectivity issue is resolved, the upgrade precheck detects the fix and resumes upgrade automatically.

BMYPC6010

IBM Fusion operator is in a `Failed` state or not found.

Severity

Error

Problem description

It indicates that the IBM Fusion operator is in a `Failed` state or not found. It can block the ability of the IBM Fusion operator or other IBM Fusion components ability to install, upgrade, or do other function.

User actions

Do the following steps to resolve the issue:

1. Check for IBM Fusion operator CSV events and conditions to find the root cause for this state. Some of the common reasons for the `Failed` state of the IBM Fusion operator are as follows:

- One of the IBM Fusion operator pod is not running. It might be due to an OOM killed error where the pod is requesting more memory than its set limit. To resolve this, update the memory limit for that deployment in CSV.
- A pod might be in a `crashloopbackoff` due to another reason. Check the pod log and events to find the root cause and take an appropriate action.
- It is possible that a `Custom Resource Definition` is missing due to an accidental deletion. In this case, contact IBM support to get the CRD YAML and create it in the cluster.

You can continue with the upgrade operation after the IBM Fusion operator is successful.

If you want to override the prechecks that are blocking the upgrade, do the following steps:

It converts the `Error` prechecks to `Warning` prechecks and triggers the upgrade.
Note: Before using the `precheck-acks` configmap to override the pre-checks, ensure that the unhealthy resource does not affect the upgrade.

1. Run the following command and export IBM Fusion namespace as an environmental variable.

   ```
   export FUSION_NS="namespace-where-fusion-is-installed"
   ```

2. Run the following command to create a `precheck-acks` configmap in the namespace where the IBM Fusion is installed.

   ```
   oc create configmap precheck-acks -n $FUSION_NS
   ```

3. In the `precheck-acks` config map, add the key `FusionOperatorStatus` and set the value to `true`.

   ```
   oc patch configmap precheck-acks -n $FUSION_NS --type merge -p '{"data": {"FusionOperatorStatus": "true"}}'
   ```

BMYPC6013

The DNS resolution for image registry failed.

Severity

Error

Problem description

It indicates that the hostname resolution specified for the image registry failed for one or more cluster nodes. It blocks the upgrade because of image pull failure. It can also cause other issues outside of the upgrade because the newly scheduled or rescheduled pods cannot access the images.

The registries can have a non-empty auth specified in the pull-secret of the `openshift-config` namespace. Check the connectivity for successful image pulls of IBM Fusion components and the OpenShift® Container Platform. Failure to address the registry access can result in an `ImagePullbackOff` error of the pod.

User actions

Do the following steps to resolve the issue:

1. Check the DNS settings for the image registry.
2. A firewall can prevent the access to the registry
3. After the DNS resolution issue is resolved, the upgrade precheck detects the fix and resumes upgrade automatically.

If you want to override the prechecks that are blocking the upgrade, do the following steps:

It converts the `Error` prechecks to `Warning` prechecks and triggers the upgrade.
Note: Before using the `precheck-acks` configmap to override the pre-checks, ensure that the unhealthy resource does not affect the upgrade.

1. Run the following command and export IBM Fusion namespace as an environmental variable.

   ```
   export FUSION_NS="namespace-where-fusion-is-installed"
   ```

2. Run the following command to create a `precheck-acks` configmap in the namespace where the IBM Fusion is installed.

   ```
   oc create configmap precheck-acks -n $FUSION_NS
   ```

3. In the `precheck-acks` configmap, add the key `DNSResolution` and set the value to `true`.

   ```
   oc patch configmap precheck-acks -n $FUSION_NS --type merge -p '{"data": {"DNSResolution": "true"}}'
   ```

BMYPC6040

Backup & Restore agent instance not found.

Severity

Error

Problem description

It indicates that there is no Backup & Restore agent instance on the cluster. This can occur if the `DataProtectionAgent` CR is deleted from the cluster.

User actions

Contact IBM support to resolve this issue.

BMYPC6042

The Subscription is missing the `CatalogSource`.

Severity

Error

Problem description

It indicates that the listed subscription is missing in the `CatalogSource`. This can occur if the `CatalogSource` is missing from the subscription CR, or if the `CatalogSource` listed in the subscription CR is missing on the cluster.

User actions

Do the following steps to resolve the issue:

1. Check the `CatalogSource` required by the subscription exists on the cluster and create the `CatalogSource` if it does not exist
2. If you want to override the prechecks that are blocking the upgrade, do the following steps:
   Note: Before using the `precheck-acks configmap` to override the prechecks, ensure that the unhealthy resource does not affect the upgrade.
   a. Run the following command and export IBM Fusion namespace as an environmental variable.

      ```
      export FUSION_NS="namespace-where-fusion-is-installed
      ```

   b. Run the following command to create a `precheck-acks configmap` in the namespace where the IBM Fusion is installed.

      ```
      oc create configmap precheck-acks -n $FUSION_NS
      ```

   c. In the `precheck-acks configmap`, add the key `BNRSubscriptionCatalogs` and set the value to true.

      ```
      oc patch configmap precheck-acks -n $FUSION_NS --type merge -p '{"data": {"BNRSubscriptionCatalogs": "true"}}'
      ```

BMYPC6044

Backup & Restore server instance is unhealthy.

Severity

Error

Problem description

It indicates that the Backup & Restore server instance is unhealthy. This can occur if one or more services listed the `DataProtectionServer` CR contain issues.

Recommended actions

Contact IBM support.

BMYPC6045

Backup & Restore agent instance is unhealthy.

Severity

Error

Problem description

It indicates that the Backup & Restore agent instance is unhealthy. This can occur if one or more services listed the `DataProtectionServer` CR contain issues.

Recommended actions

Contact IBM support.

Upgrade events and error codes

List of all the events that you might encounter during upgrade.

For Informational, Warning and Critical events, see the following:

- **BMYUP1101**
  Upgrade of Software components is in progress.
- **BMYUP1102**
  Software upgrade is complete.
- **BMYUP1103**
  Upgrade of Red Hat® OpenShift® cluster is in progress.
- **BMYUP1104**
  Red Hat OpenShift cluster upgrade is complete.
- **BMYUP1111**
  Started upgrade of the Global Data Platform service.
- **BMYUP1112**
  Completed upgrade of the Global Data Platform service.
- **BMYUP1113**
  Started upgrade of the Data Protection service.
- **BMYUP1114**
  Completed upgrade of the Data Protection service.
- **BMYUP1115**
  Started upgrade of the Data Foundation service.
- **BMYUP1116**
  Completed upgrade of the Data Foundation service.
- **BMYUP1117**
  IBM Fusion service upgrade available for the on-boarded service.
- **BMYUP1118**
  Service upgrade is available.
- **BMYUP1119**
  Service upgrade in progress.
- **BMYUP1120**
  Service upgrade is completed.
- **BMYUP1121**
  Operator Upgrade in progress.
- **BMYUP1123**
  Operator Upgrade completed.
- **BMYUP3101**
  Failed to upgrade the Global Data Platform service.
- **BMYUP3102**
  Failed to upgrade Backup & Restore service.
- **BMYUP3103**
  Failed to upgrade the Data Foundation service.
- **BMYUP3104**
  Service upgrade is failing.
- **BMYUP3105**
  Operator Upgrade Failed.
- **BMYUP3106**
  Operator upgrade failed for the service.
- **BMYPC6021**
  Compute nodes firmware upgrade is in progress.
- **BMYPC6022**
  Compute node reboot is in progress.
- **BMYPC6053**
  Switch firmware upgrade is in progress.

BMYUP1101

Upgrade of Software components is in progress.

Severity

Informational

User actions

No action is required.

BMYUP1102

Software upgrade is complete.

Severity

Informational

User actions

No action is required.

BMYUP1103

Upgrade of Red Hat® OpenShift® cluster is in progress.

Severity

Informational

User actions

No action is required.

BMYUP1104

Red Hat® OpenShift® cluster upgrade is complete.

Severity

Informational

User actions

No action is required.

BMYUP1111

Started upgrade of the Global Data Platform service.

Severity

Informational

User actions

No action is required.

BMYUP1112

Completed upgrade of the Global Data Platform service.

Severity

Informational

User actions

No action is required.

BMYUP1113

Started upgrade of the Data Protection service.

Severity

Informational

User actions

No action is required.

BMYUP1114

Completed upgrade of the Data Protection service.

Severity

Informational

User actions

No action is required.

BMYUP1115

Started upgrade of the Data Foundation service.

Severity

Informational

User actions

No action is required.

BMYUP1116

Completed upgrade of the Data Foundation service.

Severity

Informational

User actions

No action is required.

BMYUP1117

IBM Fusion service upgrade available for the on-boarded service.

Severity

Informational

User actions

No action is required.

BMYUP1118

Service upgrade is available.

Severity

Informational

User actions

No action is required.

BMYUP1119

Service upgrade in progress.

Severity

Informational

User actions

No action is required.

BMYUP1120

Service upgrade is completed.

Severity

Informational

User actions

No action is required.

BMYUP1121

Operator Upgrade in progress.

Severity

Informational

User actions

No action is required.

BMYUP1123

Operator Upgrade completed.

Severity

Informational

User actions

No action is required.

BMYUP3101

Failed to upgrade the Global Data Platform service.

Severity

Critical

User actions

Collect Global Data Platform logs and contact IBM support.

BMYUP3102

Failed to upgrade Backup & Restore service.

Severity

Critical

User actions

Do the following steps to diagnose and report the problem:

1. Ensure that you performed the before you begin steps of Upgrading IBM Fusion.
2. If the issue persists, contact IBM support.

BMYUP3103

Failed to upgrade the Data Foundation service.

Severity

Critical

User actions

Collect Data Foundation logs and contact IBM support.

BMYUP3104

Service upgrade is failing.

Severity

Critical

User actions

Do the following steps to diagnose and report the problem:

1. Go to OpenShift® Container Platform web console and check the FusionServiceInstance for the specific service and check the conditions section to find the error.
2. If the issue persists, run the following oc command, and check the status section and debug the error.

   ```
   oc get fusionserviceinstance <instance-name-of-service> -o yaml
   ```

3. If this occurs during installation or post service installation, collect logs for the corresponding service that shows as unhealthy and contact IBM support. Follow the steps to collect logs from IBM Fusion.
   - Go to Services page in IBM Fusion user interface.
   - Click the ellipses menu for the corresponding service.
   - Select the Collect logs button. It downloads all the necessary logs.

BMYUP3105

Operator Upgrade Failed.

Severity

Critical

User actions

Do the following steps to diagnose the error:

1. Go to OpenShift® Container Platform web console and check the FusionServiceInstance for the specific service and check the conditions section to find the error.
2. If the issue persists, run the following oc command to check the status section and try to debug the error.

```
oc get fusionserviceinstance <instance-name-of-service> -o yaml
```

3. If the issue still persists, do the following:
   - If this occurs during Data Cataloging upgrade, Collect Data Cataloging logs and contact IBM support.
   - If this occurs during Backup and Restore upgrade, Collect Backup and Restore logs and contact IBM support.

BMYUP3106

Operator upgrade failed for the service.

Severity

Error

Problem description

It indicates that the operator upgrade for the service is failed. You can find reason for the upgrade failure in the displayed message on the user interface.

User actions

Contact IBM support to resolve this issue.

BMYPC6021

Compute nodes firmware upgrade is in progress.

Severity

Error

Problem description

It indicates that the upgrade for the compute node firmwares are in progress. In such case, the upgrades for OpenShift® cluster, switches, and Global Data Platform service must not be triggered because the firmware upgrade requires rolling restart of OpenShift nodes.

User actions

Wait until the compute nodes silent upgrades to finish.

BMYPC6022

Compute node reboot is in progress.

Severity

Error

Problem description

It indicates that the compute node reboot is in progress. In such case, the upgrades for OpenShift® cluster, switches, and Global Data Platform service should not be triggered.

User actions

Wait until the compute node reboot to finish.

BMYPC6053

Switch firmware upgrade is in progress.

Severity

Error

Problem description

It indicates that the switch firmware upgrade is in progress. In such a case, the upgrades for OpenShift® cluster, Global Data Platform, and compute firmware must not be triggered.

User actions

Wait until the switch upgrade to finish.

Global Data Platform events and error codes

List of all the events and error codes that you might encounter while you work with storage.

- **BMYSS0001**
  IBM Storage Scale storage installation is in progress.
- **BMYSS0002**
  IBM Storage Scale storage installation failed.
- **BMYSS0003**
  IBM Storage Scale storage installation succeeded.
- **BMYSS0004**
  IBM Storage Scale storage upgrade is in progress.
- **BMYSS0005**
  IBM Storage Scale storage upgrade failed.
- **BMYSS0006**
  IBM Storage Scale storage upgrade succeeded.
- **BMYSS0007**
  IBM Storage Scale remote mount file system status is not ready.
- **BMYSS0009**
  IBM Storage Scale cluster CR status is not ready.
- **BMYSS0010**
  IBM Storage Scale operator status in not ready.
- **BMYSS0011**
  IBM Storage Scale remote mount file system installation is in progress.
- **BMYSS0012**
  IBM Storage Scale remote mount file system installation failed.
- **BMYSS0013**
  IBM Storage Scale remote mount file system installation succeeded.

BMYSS0001

IBM Storage Scale storage installation is in progress.

Severity

Informational

User actions

No action is required.

BMYSS0002

IBM Storage Scale storage installation failed.

Error conditions

The error message can be for the following conditions:

- Unable to install IBM Storage Scale operator.
- Unable to install IBM Storage Scale CSI operator.

Severity

Warning

User actions

Collect Global Data Platform logs and contact IBM support.

BMYSS0003

IBM Storage Scale storage installation succeeded.

Severity

Informational

User actions

No action is required.

BMYSS0004

IBM Storage Scale storage upgrade is in progress.

Severity

Informational

User actions

No action is required.

BMYSS0005

IBM Storage Scale storage upgrade failed.

Error conditions

The error message can be for the following conditions:

- Unable to upgrade IBM Storage Scale operator.
- Unable to upgrade IBM Storage Scale CSI operator.
- Unable to recreate the IBM Storage Scale core pods.

Severity

Warning

User actions

Collect Global Data Platform logs and contact IBM support.

BMYSS0006

IBM Storage Scale storage upgrade succeeded.

Severity

Informational

User actions

No action is required.

BMYSS0007

IBM Storage Scale remote mount file system status is not ready.

Error conditions

The error message can be for the following conditions:

- Unable to connect the remote storage cluster.
- Unable to mount filesystem on some worker node.

Severity

Warning

User actions

Collect Global Data Platform logs and contact IBM support.

BMYSS0009

IBM Storage Scale cluster CR status is not ready.

Error conditions

The error message can be for the following condition:

- Unable to configure the IBM Storage Scale cluster resources.

Severity

Warning

User actions

Collect Global Data Platform logs and contact IBM support.

BMYSS0010

IBM Storage Scale operator status in not ready.

Error conditions

The error message can be for the following conditions:

- The IBM Storage Scale operator might have crashed.
- No available node to reschedule IBM Storage Scale CSI operator pod.

Severity

Warning

User actions

Collect Global Data Platform logs and contact IBM support.

BMYSS0011

IBM Storage Scale remote mount file system installation is in progress.

Severity

Informational

User actions

No action is required.

BMYSS0012

IBM Storage Scale remote mount file system installation failed.

Error conditions

The error message can be for the following conditions:

- Unable to create filesystem.
- Unable to register client cluster on the storage cluster.

Severity

Warning

User actions

Collect Global Data Platform logs and contact IBM support.

BMYSS0013

IBM Storage Scale remote mount file system installation succeeded.

Severity

Informational

User actions

No action is required.

Data foundation events and error codes

List of all the events and error codes that you might encounter while you work with Data Foundation storage.

- **BMYSS0100**
  Data Foundation installation is in progress.
- **BMYSS0101**
  Data Foundation installation succeeded.
- **BMYSS0102**
  Data Foundation installation failed.
- **BMYSS0103**
  Data Foundation status is degraded.
- **BMYSS0104**
  Data Foundation status is healthy.
- **BMYSS0105**
  Data Foundation upgrade is in progress.
- **BMYSS0106**
  Data Foundation upgrade succeeded.
- **BMYSS0107**
  Data Foundation upgrade failed.
- **BMYSS0108**
  Data Foundation storage cluster installation is in progress.
- **BMYSS0109**
  Data Foundation storage cluster installation succeeded.
- **BMYSS0110**
  Data Foundation storage cluster status is degraded.
- **BMYSS0111**
  Data Foundation storage cluster status is healthy.

- **BMYSS0112**
  Data Foundation is scaling up automatically.
- **BMYSS0115**
  Field <fieldname> is required.
- **BMYSS0116**
  Field <fieldname> is required if authentication method is <auth_method>.
- **BMYSS0117**
  Field <fieldname> is invalid.
- **BMYSS0118**
  Client certificate and Client key need to be provided both, or neither of them to be provided.
- **BMYSS0119**
  Field <fieldname> is immutable.
- **BMYSS0120**
  KMS server is not reachable.
- **BMYSS0121**
  Failed to validate Data Foundation encryption configuration.
- **BMYSS0122**
  Data Foundation capacity is adding.
- **BMYSS0125**
  Data Foundation capacity is added.
- **BMYSS0126**
  Nodes are added to Data Foundation storage cluster successfully.
- **BMYSS0127**
  Data Foundation is auto-scaled up successfully.
- **BMYSS0128**
  Storage Class `ibm-spectrum-fusion-mgmt-sc` is not provisioned in accordance with Fusion Data Foundation guidelines.

BMYSS0100

Data Foundation installation is in progress.

Severity

Informational

User actions

No action is required.

BMYSS0101

Data Foundation installation succeeded.

Severity

Informational

User actions

No action is required.

BMYSS0102

Data Foundation installation failed.

Error conditions

The error message can be for the following conditions:

- Unable to install Data Foundation operators.
- Unable to install local storage operator.

Severity

Warning

User actions

Collect Data Foundation logs and contact IBM support.

BMYSS0103

Data Foundation status is degraded.

Error conditions

The error message can be for the following conditions:

- Some Data Foundation operators are not running.
- The CSV phase state of Data Foundation is not succeeded.

Severity

Warning

User actions

Collect Data Foundation logs and contact IBM support.

BMYSS0104

Data Foundation status is healthy.

Severity

Informational

User actions

No action is required.

BMYSS0105

Data Foundation upgrade is in progress.

Severity

Informational

User actions

No action is required.

BMYSS0106

Data Foundation upgrade succeeded.

Severity

Informational

User actions

No action is required.

BMYSS0107

Data Foundation upgrade failed.

Error conditions

The error message can be for the following condition:

- Unable to upgrade Data Foundation operators.

Severity

Warning

User actions

Collect Data Foundation logs and contact IBM support.

BMYSS0108

Data Foundation storage cluster installation is in progress.

Severity

Informational

User actions

No action is required.

BMYSS0109

Data Foundation storage cluster installation succeeded.

Severity

Informational

User actions

No action is required.

BMYSS0110

Data Foundation storage cluster status is degraded.

Error conditions

The error message can be for the following condition:

- The phase of the Data Foundation storage cluster is not ready

Severity

Warning

User actions

Collect Data Foundation logs and contact IBM support.

BMYSS0111

Data Foundation storage cluster status is healthy.

Severity

Informational

User actions

No action is required.

BMYSS0112

Data Foundation is scaling up automatically.

Severity

Informational

User actions

No action is required.

BMYSS0115

Field <fieldname> is required.

Severity

Error

User actions

Ensure that you need to put the field <fieldname>. The fields include Host, Port, Authentication Method.

BMYSS0116

Field <fieldname> is required if authentication method is <auth_method>.

Severity

Error

User actions

Do the following steps to diagnose the problem:

1. If Authentication Method is **`token`**, user need to input field **`token`**.
2. If Authentication Method is **`kubernetes`**, user need to input field **`role`**.

BMYSS0117

Field <fieldname> is invalid.

Severity

Error

User actions

Ensure that you need to put a valid value.

BMYSS0118

Client certificate and Client key need to be provided both, or neither of them to be provided.

Severity

Error

User actions

Ensure that the user must be provided with both a client certificate and a client key, or not provide either of them.

BMYSS0119

Field <fieldname> is immutable.

Severity

Error

User actions

User cannot update the field <fieldname>.

BMYSS0120

KMS server is not reachable.

Severity

Error

User actions

Ensure that you need to check connectivity of KMS server.

BMYSS0121

Failed to validate Data Foundation encryption configuration.

Error conditions

The error message can be for the following conditions:

- Encryption configuration provided is not correct.
- Error occurs when validating encryption configuration.

Severity

Warning

User actions

Do the following steps to diagnose the problem:

1. Check the event description for detailed reason and ensure that the encryption configuration provided is valid.
2. If the issue persists, collect Data Foundation logs and contact IBM support.

BMYSS0122

Data Foundation capacity is adding.

Severity

Informational

User actions

No action is required.

BMYSS0125

Data Foundation capacity is added.

Severity

Informational

User actions

No action is required.

BMYSS0126

Nodes are added to Data Foundation storage cluster successfully.

Severity

Informational

User actions

No action is required.

BMYSS0127

Data Foundation is auto-scaled up successfully.

Severity

Informational

User actions

No action is required.

BMYSS0128

Storage Class **ibm-spectrum-fusion-mgmt-sc** is not provisioned in accordance with Fusion Data Foundation guidelines.

Severity

Critical

User actions

To resolve this critical event, manually delete the storage claim **ibm-spectrum-fusion-mgmt-sc**.

1. Check for PVCs that are using the storage class **ibm-spectrum-fusion-mgmt-sc**.

   ```
   oc get pvc --all-namespaces -o json | jq '.items[] | select(.spec.storageClassName == "ibm-spectrum-fusion-mgmt-sc") | {namespace: .metadata.namespace, name: .metadata.name, storageClass: .spec.storageClassName}'
   ```

2. Migrate PVCs (if any) from **ibm-spectrum-fusion-mgmt-sc** storage class to **ocs-storagecluster-cephfs** storage class.
   If there are any application PVCs using this storage class, plan to migrate them to the **ocs-storagecluster-cephfs** storage class. Go through the Red Hat documentation for guidance on migrating PVCs.
   Note: If any service (Backup & Restore,IBM Data Cataloging, or CAS) is installed on the cluster and is using the **ibm-spectrum-fusion-mgmt-sc** storage class, contact IBM support.
3. Delete the storage claim.
   After all PVCs are successfully migrated and none are using the **ibm-spectrum-fusion-mgmt-sc** storage class, run the following command to manually delete the storage claim:

   ```
   oc delete storageclaim ibm-spectrum-fusion-mgmt-sc
   ```

4. Verify deletion of the storage class.
   After you delete the storage claim, run the following command to ensure that the storage class does not exist:

```
oc get storageclass | grep ibm-spectrum-fusion-mgmt-sc
```

IBM Data Cataloging events and error codes

List of all the events that you might encounter while you work with IBM Data Cataloging for the IBM Fusion.

For Informational, Warning, and Critical events, see the following:

- **BMYDC0001**
  IBM Data Cataloging service is healthy.
- **BMYDC0002**
  IBM Data Cataloging service health is degraded.
- **BMYDC0003**
  IBM Data Cataloging service workload resources are ready.
- **BMYDC0004**
  IBM Data Cataloging service workload resources are not ready.
- **BMYDC0005**
  IBM Data Cataloging service API is available.
- **BMYDC0006**
  IBM Data Cataloging service API is not completely available.
- **BMYDC0100**
  IBM Data Cataloging service installation in progress.
- **BMYDC0101**
  IBM Data Cataloging service installation succeeded.
- **BMYDC0102**
  IBM Data Cataloging service installation failing.
- **BMYDC0103**
  IBM Data Cataloging service is already installed.
- **BMYDC0104**
  IBM Data Cataloging service upgrade in progress.
- **BMYDC0105**
  IBM Data Cataloging service upgrade succeeded.
- **BMYDC0106**
  IBM Data Cataloging service upgrade failing.
- **BMYDC0107**
  IBM Data Cataloging service is not ready for upgrade.
- **BMYDC0108**
  New version is available for IBM Data Cataloging service upgrade.
- **BMYDC0109**
  No new versions available for IBM Data Cataloging service upgrade.
- **BMYDC0110**
  IBM Data Cataloging service prerequisites are ready.
- **BMYDC0113**
  IBM Data Cataloging service Kafka instances are ready.
- **BMYDC0114**
  IBM Data Cataloging service Kafka instances are not ready.
- **BMYDC0115**
  IBM Data Cataloging service Db2 is ready.
- **BMYDC0116**
  IBM Data Cataloging service Db2 is not ready.
- **BMYDC0117**
  IBM Data Cataloging service license was not accepted.
- **BMYDC0118**
  Cluster API that is required by the IBM Data Cataloging service is not available.

BMYDC0001

IBM Data Cataloging service is healthy.

Severity

Informational

User actions

No action is required.

BMYDC0002

IBM Data Cataloging service health is degraded.

Severity

Warning

User actions

Do the following steps to diagnose and report the problem:

1. Run the following command and check whether if there is any issue with the specific components listed.

```
oc -n ibm-data-cataloging get deployment -l 'component=discover,role in (api,connmgr,db2whrest,policyengine,isd-proxy,ui)'
```

```
Expected output
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
isd-api            1/1     1            1           6d11h
isd-connmgr        1/1     1            1           6d11h
isd-db2whrest      1/1     1            1           6d11h
isd-policyengine   1/1     1            1           6d11h
isd-proxy          1/1     1            1           6d11h
isd-ui-backend     1/1     1            1           6d11h
isd-ui-frontend    1/1     1            1           6d11h
```

2. If you find any issue with the specific components listed, collect the Data cataloging logs and contact IBM support.

BMYDC0003

IBM Data Cataloging service workload resources are ready.

Severity

Informational

User actions

No action is required.

BMYDC0004

IBM Data Cataloging service workload resources are not ready.

Severity

Warning

User actions

Do the following steps to diagnose and report the problem:

1. Run the following command and check whether if there is any issue with the specific components listed.

```
oc -n ibm-data-cataloging get deployment -l 'component=discover'
```

```
Expected output
NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
isd-api                     1/1     1            1           6d11h
isd-auth                    1/1     1            1           6d11h
isd-backup-restore          1/1     1            1           6d11h
isd-connmgr                 1/1     1            1           6d11h
isd-consumer-ceph-le        10/10   10           10          6d11h
isd-consumer-cos-le         10/10   10           10          6d11h
isd-consumer-cos-scan       10/10   10           10          6d11h
isd-consumer-file-scan      10/10   10           10          6d11h
isd-consumer-protect-scan   10/10   10           10          6d11h
isd-consumer-scale-le       10/10   10           10          6d11h
isd-consumer-scale-scan     10/10   10           10          6d11h
isd-contentsearchagent      1/1     1            1           6d11h
isd-db2whrest               1/1     1            1           6d11h
isd-importtags              1/1     1            1           6d11h
isd-keystone                1/1     1            1           6d11h
isd-policyengine            1/1     1            1           6d11h
isd-producer-ceph-le        1/1     1            1           6d11h
isd-producer-cos-le         1/1     1            1           6d11h
isd-producer-cos-scan       1/1     1            1           6d11h
isd-producer-file-scan      1/1     1            1           6d11h
isd-producer-protect-scan   1/1     1            1           6d11h
isd-producer-scale-le       1/1     1            1           6d11h
isd-producer-scale-scan     1/1     1            1           6d11h
isd-proxy                   1/1     1            1           6d11h
isd-scaleafmdatamover       1/1     1            1           6d11h
```

```
isd-scaleilmdatamover       1/1     1            1           6d11h
isd-sdmonitor               1/1     1            1           6d11h
isd-tikaserver              1/1     1            1           6d11h
isd-ui-backend              1/1     1            1           6d11h
isd-ui-frontend             1/1     1            1           6d11h
isd-wkcconnector            1/1     1            1           6d11h
```

2. If you find any issue with the specific components listed, collect the Data cataloging logs and contact IBM support.

BMYDC0005

IBM Data Cataloging service API is available.

Severity

Informational

User actions

No action is required.

BMYDC0006

IBM Data Cataloging service API is not completely available.

Severity

Warning

User actions

Do the following steps to diagnose and report the problem:

1. Run the following command and check whether if there is any issue with the specific components listed.

```
oc -n ibm-data-cataloging get deployment -l 'component=discover,role in (api,connmgr,db2whrest,policyengine,isd-proxy,ui)'
```

```
Expected output
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
isd-api            1/1     1            1           6d11h
isd-connmgr        1/1     1            1           6d11h
isd-db2whrest      1/1     1            1           6d11h
isd-policyengine   1/1     1            1           6d11h
isd-proxy          1/1     1            1           6d11h
isd-ui-backend     1/1     1            1           6d11h
isd-ui-frontend    1/1     1            1           6d11h
```

2. If you find any issue with the specific components listed, collect the Data cataloging logs and contact IBM support.

BMYDC0100

IBM Data Cataloging service installation in progress.

Severity

Informational

User actions

No action is required.

BMYDC0101

IBM Data Cataloging service installation succeeded.

Severity

Informational

User actions

No action is required.

BMYDC0102

IBM Data Cataloging service installation failing.

Severity

Critical

User actions

Do the following steps to diagnose and report the problem:

1. Run the following commands and collect the `ibm_data_cataloging_fsd.yaml, ibm_data_cataloging_fsi.yaml, ibm_data_cataloging_cr.yaml and ibm_data_cataloging_operator_logs.yaml` files information.

   ```
   oc -n ibm-spectrum-fusion-ns get fusionservicedefinition data-cataloging-service-definition -o yaml > ibm_data_cataloging_fsd.yaml
   oc -n ibm-spectrum-fusion-ns get fusionserviceinstance data-cataloging-service-instance -o yaml > ibm_data_cataloging_fsi.yaml
   oc -n ibm-data-cataloging get isd -o yaml > ibm_data_cataloging_cr.yaml
   oc -n ibm-data-cataloging logs deployment/spectrum-discover-operator > ibm_data_cataloging_operator_logs.yaml
   ```

2. Contact IBM support and provide the `ibm_data_cataloging_fsd.yaml, ibm_data_cataloging_fsi.yaml, ibm_data_cataloging_cr.yaml and ibm_data_cataloging_operator_logs.yaml` files information.

BMYDC0103

IBM Data Cataloging service is already installed.

Severity

Informational

User actions

No action is required.

BMYDC0104

IBM Data Cataloging service upgrade in progress.

Severity

Informational

User actions

No action is required.

BMYDC0105

IBM Data Cataloging service upgrade succeeded.

Severity

Informational

User actions

No action is required.

BMYDC0106

IBM Data Cataloging service upgrade failing.

Severity

Critical

User actions

Do the following steps to diagnose and report the problem:

1. Run the following commands and collect the `ibm_data_cataloging_fsd.yaml, ibm_data_cataloging_fsi.yaml, ibm_data_cataloging_cr.yaml and ibm_data_cataloging_operator_logs.yaml` files information.

   ```
   oc -n ibm-spectrum-fusion-ns get fusionservicedefinition data-cataloging-service-definition -o yaml > ibm_data_cataloging_fsd.yaml
   oc -n ibm-spectrum-fusion-ns get fusionserviceinstance data-cataloging-service-instance -o yaml > ibm_data_cataloging_fsi.yaml
   oc -n ibm-data-cataloging get isd -o yaml > ibm_data_cataloging_cr.yaml
   oc -n ibm-data-cataloging logs deployment/spectrum-discover-operator > ibm_data_cataloging_operator_logs.yaml
   ```

2. Contact IBM support and provide the `ibm_data_cataloging_fsd.yaml, ibm_data_cataloging_fsi.yaml, ibm_data_cataloging_cr.yaml and ibm_data_cataloging_operator_logs.yaml` files information.

BMYDC0107

IBM Data Cataloging service is not ready for upgrade.

Severity

Informational

User actions

No action is required.

BMYDC0108

New version is available for IBM Data Cataloging service upgrade.

Severity

Informational

User actions

No action is required.

BMYDC0109

No new versions available for IBM Data Cataloging service upgrade.

Severity

Informational

User actions

No action is required.

BMYDC0110

IBM Data Cataloging service prerequisites are ready.

Severity

Informational

User actions

No action is required.

BMYDC0113

IBM Data Cataloging service Kafka instances are ready.

Severity

Informational

User actions

No action is required.

BMYDC0114

IBM Data Cataloging service Kafka instances are not ready.

Severity

Warning

User actions

Do the following steps to diagnose and report the problem:

1. Run the following command and check whether if there is any issue with the specific components listed.

   ```
   oc -n ibm-data-cataloging get kafka isd-ssl isd-sasl
   ```

   ```
   Expected output
   NAME       DESIRED KAFKA REPLICAS   DESIRED ZK REPLICAS   READY   WARNINGS
   isd-ssl    1                        1                     True
   isd-sasl   1                        1                     True
   ```

2. If you find any issue with the specific components listed, collect the Data cataloging logs and contact IBM support.

BMYDC0115

IBM Data Cataloging service Db2 is ready.

Severity

Informational

User actions

No action is required.

BMYDC0116

IBM Data Cataloging service Db2 is not ready.

Severity

Warning

User actions

Do the following steps to diagnose and report the problem:

1. Run the following command and check whether if there is any issue with the specific components listed.

```
oc -n ibm-data-cataloging get db2u

Expected output
NAME    STATE   MAINTENANCESTATE   AGE
isd     Ready   None               6d15h
```

2. If you find any issue with the specific components listed, collect the Data cataloging logs and contact IBM support.

BMYDC0117

IBM Data Cataloging service license was not accepted.

Severity

Critical

User actions

Do the following steps to diagnose and report the problem:

1. This is not expected, by default IBM Data Cataloging license is accepted.
2. Run the following command to check the IBM Data Cataloging.

```
ISD_INSTANCE=$(oc -n ibm-data-cataloging get isd -o name | head -n1)
oc -n ibm-data-cataloging patch $ISD_INSTANCE --type='merge' -p '{"spec":{"license":{"accept":true}}}'

Expected output
spectrumdiscover.spectrum-discover.ibm.com/data-cataloging-service-instance patched
```

BMYDC0118

Cluster API that is required by the IBM Data Cataloging service is not available.

Severity

Critical

User actions

Do the following steps to diagnose and report the problem:

1. To avoid installations outside of most OpenShift® clusters, this event is only triggered when a cluster API that should be available by default in OpenShift is not present. The action to take is to attempt the install on any supported OpenShift version.
2. Attempt installation in an OpenShift cluster running any of the supported versions (4.14, 4.15, 4.16, 4.17, and 4.18).

Backup & Restore events and error codes

List of all the events that you might encounter during Backup & Restore for the IBM Fusion.

For Informational, Warning, and Critical events, see the following:

- **BMYBR0001**
  The job has not progressed in the last hour.
- **BMYBR0002**
  The job was not started because of an already running job.
- **BMYBR0003**
  There was an error when processing the job in the service.
- **BMYBR0009**
  There was an error when processing the job in the service.
- **BMYDP1146**
  Policy assignment is created initial backup for an application.
- **BMYDP1163**
  Completed backup for application from backup policy.

BMYBR0001

The job has not progressed in the last hour.

Severity

Warning

User actions

Collect Backup and Restore logs and contact IBM support.

BMYBR0002

The job was not started because of an already running job.

Severity

Informational

User actions

No action is required.

BMYBR0003

There was an error when processing the job in the service.

Severity

Critical

User actions

Collect Backup and Restore logs and contact IBM support.

BMYBR0009

There was an error when processing the job in the service.

Severity

Critical

User actions

Error condition: Backup fails with the following FailedValidation error

The backup fails with the following FailedValidation error and the namespace contains encrypted Ceph RBD PVC in Block volume mode:

```
BMYBR0009
There was an error when processing the job in the Transaction Manager service. The underlying error was:
Found unsupported PVC: encrypted Ceph RBD PVC in Block volume mode cannot be safely restored <namespace/pvc>. Only Filesystem
```

Delete encrypted Ceph RBD PVC in Block volume mode from the namespace. Use Filesystem volume mode for encrypted Ceph RBD PVCs.
Note: LUKS (Linux Unified Key Setup) encrypted Ceph RBD PVC with Block volume mode are not supported. LUKS (Linux Unified Key Setup) encrypted Ceph RBD PVC with Filesystem volume mode are supported.

In general for these processing errors, collect Backup and restore logs and contact IBM support.

BMYDP1146

Policy assignment is created initial backup for an application.

Severity

Informational

User actions

No action is required.

BMYDP1163

Completed backup for application from backup policy.

Severity

Informational

User actions

No action is required.

Connection services events and error codes

List of all the events that you might encounter during connection service for the IBM Fusion.

Connection service is a common service for the Backup & Restore.

For Informational, Warning and Critical events, see the following:

- **BMYCS100**
  Create a bootstrap secret in progress.
- **BMYCS101**
  Create bootstrap secret completed.
- **BMYCS102**
  Create a kubeconfig secret in progress.
- **BMYCS103**
  Create kubeconfig secret completed.
- **BMYCS300**
  Init secret has the same API endpoint as the local cluster.
- **BMYCS301**
  The Bootstrap token in the init secret is not correct or expired.
- **BMYCS302**
  Failed to get infrastructure provider of the remote cluster.
- **BMYCS303**
  Failed to get CA certificate from the remote cluster.
- **BMYCS304**
  The CA certificate of the remote cluster is not correct.
- **BMYCS305**
  Failed to get the cluster name of the remote cluster.
- **BMYCS306**
  Failed to wait for the client configuration of the remote cluster to be ready.

BMYCS100

Create a bootstrap secret in progress.

Severity

Informational

User actions

No action is required.

BMYCS101

Create bootstrap secret completed.

Severity

Informational

User actions

No action is required.

BMYCS102

Create a kubeconfig secret in progress.

Severity

Informational

User actions

No action is required.

BMYCS103

Create kubeconfig secret completed.

Severity

Informational

User actions

No action is required.

BMYCS300

Init secret has the same API endpoint as the local cluster.

Severity

Critical

User actions

Do the following steps to resolve the error:

1. Edit the init secret with the right remote cluster API endpoint.
2. If the issue persists, then collect Storage logs and contact IBM support.

BMYCS301

The Bootstrap token in the init secret is not correct or expired.

Severity

Critical

User actions

Do the following steps to resolve the error:

1. Re-generate the bootstrap token and replace it in the init secret.
2. If the issue persists, then collect Storage logs and contact IBM support.

BMYCS302

Failed to get infrastructure provider of the remote cluster.

Severity

Critical

User actions

Collect Storage logs and contact IBM support.

BMYCS303

Failed to get CA certificate from the remote cluster.

Severity

Critical

User actions

Do the following steps to resolve the error:

1. Check if the configmap **`kube-root-ca.crt`** in namespace kube-public has CA certs in the remote cluster.
2. If the issue persists, then collect Storage logs and contact IBM support.

BMYCS304

The CA certificate of the remote cluster is not correct.

Severity

Critical

User actions

Do the following steps to resolve the error:

1. Check if the configmap **`kube-root-ca.crt`** in namespace kube-public has the right CA. If not, put the right CA in this configmap.
2. If the issue persists, then collect Storage logs and contact IBM support.

BMYCS305

Failed to get the cluster name of the remote cluster.

Severity

Critical

User actions

Collect Storage logs and contact IBM support.

BMYCS306

Failed to wait for the client configuration of the remote cluster to be ready.

Severity

Critical

User actions

Collect Storage logs and contact IBM support.

Content-Aware Storage events and error codes

List of all the events that you might encounter while you work with CAS for the IBM Fusion.

For Informational, Warning, and Critical events, see the following:

- BMYCA0201
  `KAFKAInstallFail`
- BMYCA0204
  `QueryServiceInstallFail`
- BMYCA0216
  `DProcCRAbsent`
- BMYCA0217
  `PipelineInstanceRetrivalFailed`
- BMYCA0225
  `KAFKAApplyYamlFailed`
- BMYCA0227
  `KAFKAClusterDeploymentFailed`
- BMYCA0228
  `KAFKAClusterNotHealthy`
- BMYCA0229
  `KAFKAInstanceRetrivalFailed`
- BMYCA0230
  `KAFKAClusterStatusRetrievalFailed`
- BMYCA0231
  `KAFKAInstanceNotReady`
- BMYCA0232
  `ParadeDBClusterRetrivalFailed`
- BMYCA0233
  `ParadeDBClusterDeploymentFailed`
- BMYCA0234
  `ParadeDBClusterNotHealthy`
- BMYCA0235
  `ParadeDBPoolerInstanceRetrivalFailed`
- BMYCA0243
  `ParadeDBConnectionFail`
- BMYCA0244
  `QueryServicePodFailed`
- BMYCA0245
  `QueryServiceNotHealthy`
- BMYCA0258
  `GetUsersFail`
- BMYCA0259
  `ValidateUsersFail`
- BMYCA0260
  `GetGroupsFail`
- BMYCA0261
  `ValidateGroupsFail`
- BMYCA0263
  `DBConnectionFail`
- BMYCA0264
  `DBPopulationFail`
- BMYCA0266
  `DeleteEntriesOfUsersAndGroupFromACLTableFail`
- BMYCA0267
  `DeleteEntriesOfDomainsFromACLTableFail`
- BMYCA0268
  `ClusterWatchDisableFail`

BMYCA0201

`KAFKAInstallFail`

Severity

Critical

Problem description

It indicates that Kafka has failed to deploy.

User actions

Follow the steps to resolve the issue:

1. Review the `cas-operator` logs to identify the cause of the Kafka installation failure.
2. Collect CAS logs and check whether you find any errors that are related to CAS operator.
3. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0204

`QueryServiceInstallFail`

Severity

Critical

Problem description

It indicates that the query service failed to install correctly.

User actions

Follow the steps to resolve the issue:

1. Check whether `query-service` pods are running in the `ibm-cas` namespace.
2. Check whether image is being correctly pulled. If not, check the `icr.io` pull-secret.
3. Check logs in `ibm-cas-operator-controller` pod for query service errors.
4. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0216

`DProcCRAbsent`

Severity

Critical

Problem description

The `DocumentProcessor` CR is missing.

User actions

Review the `cas-operator` logs to identify the cause of the CR creation failure.

BMYCA0217

`PipelineInstanceRetrivalFailed`

Severity

Critical

Problem description

The pipeline instance retrieval has failed.

User actions

Review the `cas-operator` logs to identify the cause of the pipeline instance retrieval failure.

BMYCA0225

`KAFKAApplyYamlFailed`

Severity

Critical

Problem description

It indicates that the Kafka installation has failed.

User actions

Follow the steps to resolve the issue:

1. Check whether the Kafka operator is installed successfully.
2. Check whether any errors or waring messages in CAS installation.
3. Collect CAS logs and check whether you find any errors that are related to CAS operator.
4. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0227

KAFKAClusterDeploymentFailed

Severity

Critical

Problem description

It indicates that the Kafka installation has failed.

User actions

Follow the steps to resolve the issue:

1. Check whether the Kafka operator is installed successfully.
2. Check whether any errors or waring messages in CAS installation.
3. Collect CAS logs and check whether you find any errors that are related to CAS operator.
4. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0228

KAFKAClusterNotHealthy

Severity

Low

Problem description

It indicates that the Kafka cluster is not in a healthy state.

User actions

Follow the steps to resolve the issue:

1. Wait until the Kafka instance is present to let the log info fix by itself.
2. Check whether the Kafka operator is installed successfully.
3. Check whether any errors or warning messages in CAS installation.
4. Collect CAS logs and check whether you find any errors that are related to CAS operator.
5. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0229

KAFKAInstanceRetrivalFailed

Severity

Low

Problem description

It indicates that the Kafka instance is not present or is being installed.

User actions

Follow the steps to resolve the issue:

1. Wait until the Kafka instance is present to let the log info fix by itself.
2. Check whether the streams for Apache Kafka operator are installed successfully.
3. Collect CAS logs and check whether you find any errors that are related to CAS operator.
4. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0230

KAFKAClusterStatusRetrievalFailed

Severity

Low

Problem description

It indicates that the Kafka cluster status has failed to retrieve.

User actions

Follow the steps to resolve the issue:

1. Wait until the Kafka instance is present to let the log info fix by itself.
2. Check whether the streams for Apache Kafka operator are installed successfully.
3. Collect CAS logs and check whether you find any errors that are related to CAS operator.
4. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0231

KAFKAInstanceNotReady

Severity

Low

Problem description

It indicates that the Kafka instance is not in the ready state.

User actions

Follow the steps to resolve the issue:

1. Wait until the Kafka instance is present to let the log info fix by itself.
2. Check whether the streams for Apache Kafka operator are installed successfully.
3. Collect CAS logs and check whether you find any errors that are related to CAS operator.
4. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0232

ParadeDBClusterRetrivalFailed

Severity

Critical

Problem description

It indicates that the **ParadeDB** cluster has failed to retrieve.

User actions

Follow the steps to resolve the issue:

1. Check whether the CNPG operator is installed successfully.
2. Check whether the `RemoteFilesystem` status is connected.
3. Check whether all PVCs in the `ibm-cas` namespace in a bound state.
4. Collect CAS logs and check whether you find any errors that are related to CAS operator.
5. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0233

`ParadeDBClusterDeploymentFailed`

Severity

Critical

Problem description

It indicates that the Parade DB Cluster has failed to deploy.

User actions

Follow the steps to resolve the issue:

1. Check whether the CloudNativePG operator is installed successfully.
2. Check whether the Parade DB pods are running in the `ibm-cas` namespace.
3. Collect CAS logs and check whether you find any errors that are related to CAS operator and CloudNativePG operator.
4. Check the logs whether there is any other related error code.
5. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0234

`ParadeDBClusterNotHealthy`

Severity

Critical

Problem description

It indicates that the **`ParadeDB`** cluster is not in a healthy state.

User actions

Follow the steps to resolve the issue:

1. Check whether the CNPG operator is installed successfully.
2. Check whether the `RemoteFilesystem` status is connected.
3. Check whether all PVCs in the `ibm-cas` namespace in a bound state.
4. Collect CAS logs and check whether you find any errors that are related to CAS operator.
5. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0235

`ParadeDBPoolerInstanceRetrivalFailed`

Severity

Critical

Problem description

It indicates that the **`ParadeDB`** pooler instance has failed to retrieve.

User actions

Follow the steps to resolve the issue:

1. Check whether the CNPG operator is installed successfully.
2. Check whether the **RemoteFilesystem** status is connected.
3. Check whether all PVCs in the **ibm-cas** namespace in a bound state.
4. Collect CAS logs and check whether you find any errors that are related to CAS operator.
5. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0243

ParadeDBConnectionFail

Severity

Critical

Problem description

It indicates that the **ParadeDB** connection has failed.

User actions

Follow the steps to resolve the issue:

1. Check whether the CNPG operator is installed successfully.
2. Check whether the **RemoteFilesystem** status is connected.
3. Check whether all PVCs in the **ibm-cas** namespace in a bound state.
4. Collect CAS logs and check whether you find any errors that are related to CAS operator.
5. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0244

QueryServicePodFailed

Severity

Critical

Problem description

It indicates that the query service pod fails to maintain a running state.

User actions

Follow the steps to resolve the issue:

1. Check whether **query-service** pods are running in the **ibm-cas** namespace.
2. Check whether image is being correctly pulled. If not, check the **icr.io** pull-secret.
3. Check logs in **ibm-cas-operator-controller** pod in the **ibm-cas** namespace for query service errors.
4. Check logs in **query-service** pod in the **ibm-cas** namespace for query service errors.
5. Check for errors in the **query-service** deployment in the **ibm-cas** namespace.
6. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0245

QueryServiceNotHealthy

Severity

Critical

Problem description

It indicates that the query service is missing one or more expected resources.

User actions

Follow the steps to resolve the issue:

1. Check whether `query-service` pods are running in the `ibm-cas` namespace.
2. Check whether image is being correctly pulled. If not, check the `icr.io` pull-secret.
3. Check logs in `ibm-cas-operator-controller` pod in the `ibm-cas` namespace for query service errors.
4. Check logs in `query-service` pod in the `ibm-cas` namespace for query service errors.
5. Check for errors in the `query-service` deployment in the `ibm-cas` namespace.
6. Check for the presence of the `ibm-cas` route in the `ibm-cas` namespace.
7. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0258

`GetUsersFail`

Severity

Critical

Problem description

It indicates that the CAS service failed to retrieve users from the identity provider (Red Hat OpenShift or external identity provider depending on configuration).

User actions

Follow the steps to resolve the issue:

1. Collect CAS logs and check CAS operator pod logs to identify the cause of the failure while retrieving users from the identity provider.
2. If external identity provider is configured, validate identity provider configuration including `issuerURL`, and `client_id` and `client_secret` parameters defined in the secret.
3. Ensure that the identity provider is up and accessible to the cluster.
4. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0259

`ValidateUsersFail`

Severity

Critical

Problem description

It indicates that one or more users who are specified in the `CasResourceAccessControl` (CRAC) custom resource (CR) are not present in Red Hat OpenShift or the configured external identity providers depending on identity provider configuration.

User actions

Follow the steps to resolve the issue:

1. Check the `status.conditions` field of the CRAC CR for the `ValidatedUsers` condition, which lists the users that failed validation.
2. Ensure that the users are correctly defined in the CRAC CR.
3. Verify that the specified users exist either in the Red Hat OpenShift cluster or the external identity provider.
4. If the issue persists, collect relevant logs and contact IBM support for assistance.

BMYCA0260

`GetGroupsFail`

Severity

Critical

Problem description

It indicates that the CAS service failed to retrieve groups from the identity provider (Red Hat OpenShift or external identity provider depending on configuration).

User actions

Follow the steps to resolve the issue:

1. Collect CAS logs and check CAS operator pod logs to identify the cause of the failure while retrieving groups from the identity provider.
2. If the external identity provider is configured, validate identity provider configuration including **`issuerURL`**, and **`client_id`** and **`client_secret`** parameters defined in the secret.
3. Ensure that the identity provider is up and accessible to the cluster.
4. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0261

`ValidateGroupsFail`

Severity

Critical

Problem description

It indicates that one or more groups that are specified in the **`CasResourceAccessControl`** (CRAC) custom resource (CR) are not present in Red Hat OpenShift or the configured external identity providers depending on identity provider configuration.

User actions

Follow the steps to resolve the issue:

1. Check the **`status.conditions`** field of the CRAC CR for the **`ValidatedGroups`** condition, which lists the groups that failed validation.
2. Ensure that the groups are correctly defined in the CRAC CR.
3. Verify that the specified groups exist either in the Red Hat OpenShift cluster or the external identity provider.
4. If the issue persists, collect relevant logs and contact IBM support for assistance.

BMYCA0263

`DBConnectionFail`

Severity

Critical

Problem description

It indicates that the database (DB) connection has failed.

User actions

Follow the steps to resolve the issue:

1. Check whether the CNPG operator is installed successfully.
2. Check whether the **`RemoteFilesystem`** status is connected.
3. Check whether all PVCs in the **`ibm-cas`** namespace in a bound state.
4. Collect CNPG operator logs.
5. Collect CAS logs and check whether you find any errors that are related to CAS operator.
6. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0264

`DBPopulationFail`

Severity

Critical

Problem description

It indicates that the database (DB) population has failed.

User actions

Follow the steps to resolve the issue:

1. Check whether the CNPG operator is installed successfully.
2. Check whether the `RemoteFilesystem` status is connected.
3. Check whether all PVCs in the `ibm-cas` namespace in a bound state.
4. Collect CNPG operator logs.
5. Collect CAS logs and check whether you find any errors that are related to CAS operator.
6. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0266

`DeleteEntriesOfUsersAndGroupFromACLTableFail`

Severity

Critical

Problem description

It indicates that the deletion of user and group entries from the Access Control List (ACL) database table failed.

User actions

Follow the steps to resolve the issue:

1. Review the `cas-operator` logs to identify the cause of the failure when deleting user and group entries from the Access Control List (ACL) table.
2. Collect CAS logs and check whether you find any errors that are related to CAS operator.
3. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0267

`DeleteEntriesOfDomainsFromACLTableFail`

Severity

Critical

Problem description

It indicates that the deletion of domain entries from the Access Control List (ACL) database table failed.

User actions

Follow the steps to resolve the issue:

1. Review the `cas-operator` logs to identify the cause of the failure when deleting domain entries from the Access Control List (ACL) table.
2. Collect CAS logs and check whether you find any errors that are related to CAS operator.
3. Take appropriate action based on the error or contact IBM support to resolve the issue.

BMYCA0268

`ClusterWatchDisableFail`

Severity

Critical

Problem description

It indicates that the Watch Folder Deletion of associated data source has failed.

User actions

Follow the steps to resolve the issue:

1. Check whether the CAS operator is installed successfully.
2. Check whether the Remote Scale Cluster is connected.
3. Collect CAS logs and check whether you find any errors that are related to CAS operator.
4. Take appropriate action based on the error or contact IBM support to resolve the issue.

Serviceability events and error codes

List of all the events that you might encounter that related to serviceability.

For Informational, Warning, and Critical events, see the following:

- **BMYLC0001**
  Remaining storage space is less than 30 percent.
- **BMYLC0002**
  Remaining storage space is less than 10 percent.

BMYLC0001

Remaining storage space is less than 30 percent.

Severity

Warning

User actions

Do the following steps to resolve the error:

1. Delete the unnecessary logs through the IBM Fusion user interface to get storage space. For more information, see Delete a log package.
2. If the issue persists, then contact IBM support.

BMYLC0002

Remaining storage space is less than 10 percent.

Severity

Critical

User actions

Do the following steps to resolve the error:

1. Delete the unnecessary logs through the IBM Fusion user interface to get storage space. For more information, see Delete a log package.
2. If the issue persists, then contact IBM support.

Troubleshooting issues in IBM Fusion

This topic covers the most common issues and their workarounds, which you might encounter while you work with the IBM Fusion software.

Topic	Troubleshooting reference link
IBM Fusion install and upgrade	Installation and upgrade issues
Fusion services installation issues	• Fusion Data Foundation service- Fusion Data Foundation service install and upgrade issues • Global Data Platform service- Global Data Platform service install and upgrade issues • Backup & Restore service- Backup & Restore service installation and upgrade issues • IBM Data Cataloging service- IBM Data Cataloging service installation and upgrade issues
Global Data Platform Storage issues	Troubleshooting Global Data Platform issues
Fusion Data Foundation Storage issues	Fusion Data Foundation service error scenarios
Data protection	Troubleshooting Backup & Restore service issues
User interface issues	IBM Fusion user interface issues
Data cataloging	IBM Fusion IBM Data Cataloging known issues
Multi-cluster IBM Fusion HCI System using Hosted Control Plane	Troubleshooting issues and known limitations in Hosted Control Plane cluster
Virtualization	Virtualization issues

- **Connection setup after OpenShift Container Platform cluster recovery**
  The OpenShift® Container Platform cluster can have problems and become unusable. After you recover the cluster, rejoin the connections.
- **Network policies**
  This section provides the configuration to troubleshoot network policies.
- **Virtualization issues**
  This topic covers issues related to virtualization and their resolution.

Connection setup after OpenShift Container Platform cluster recovery

The OpenShift® Container Platform cluster can have problems and become unusable. After you recover the cluster, rejoin the connections.

Disaster recovery

- Two clusters must be connected to failover and failback applications
- Reconnect to another cluster

Backup & Restore

- One hub cluster and more than one spoke clusters can be connected to manage application backups in multi clusters.
- Reconnect to the hub or spoke

If the cluster recovers before the expiration time, then no action is needed as the cluster rejoins the connection automatically. If the cluster recovers after the client cert expires, then clean the connection and setup again to rejoin the recovered cluster. You can get the client cert effective time from the connection CR status, for example:

```
oc get connection <connection_name> -n ibm-spectrum-fusion-ns
```

```
  - lastTransitionTime: '2024-01-03T10:23:20Z'
    message: >-
      client certificate rotated starting from 2024-01-03 10:18:31 +0000 UTC
      to 2024-01-19 13:12:00 +0000 UTC
    reason: ClientCertificateUpdated
    status: 'True'
    type: ClusterCertificateRotated
```

Clean the connection and setup the connection between cluster-a and cluster-b again.

1. Clean the connection:
   Delete the connection CR in both cluster-a and cluster-b clusters:
   - In cluster-a

     ```
     oc delete connection <connection_cr_with_cluster-b_endpoint_in_spec> -n ibm-spectrum-fusion-ns
     ```

   - In cluster-b

     ```
     oc delete connection <connection_cr_with_cluster-a_endpoint_in_spec> -n ibm-spectrum-fusion-ns
     ```

2. Set up the connection between clusters again.
   a. Get the bootstrap token in cluster-a:

      ```
      kubectl create token isf-application-operator-cluster-bootstrap -n <namespace of isf-application-operator>
      ```

   b. Create connection CR and `init` secret in cluster-b with the bootstrap token and API endpoint of cluster-a.
      For example, create `init` secret in cluster-b:

      ```
      apiVersion: v1
      kind: Secret
      metadata:
        name: init-<cluster-a>
        namespace: ibm-spectrum-fusion-ns
      stringData:
        apiserver: <cluster-a api endpoint>
        bootstrapToken: <token generated in step 2.1>
      ```

3. Create connection CR with this `init` secret in `spec` in cluster-b:

   ```
   apiVersion: application.isf.ibm.com/v1
   kind: Connection
   metadata:
     name: <connection-name>
     namespace: ibm-spectrum-fusion-ns
   spec:
     remoteCluster
       apiEndpoint: <cluster-a api endpoint>
       initSecretName: init-<cluster-a>
   ```

Network policies

This section provides the configuration to troubleshoot network policies.

Benefits of network policy audit logging

The network policies that are shipped with IBM Fusion enforce traffic rules silently, allowing or denying connections based on predefined policies. When an application fails to connect, the lack of visibility into these decisions makes it difficult to determine whether the issue is caused by a network misconfiguration, a code defect, or a security policy violation.

Enabling network policy audit logging provides the following benefits:

- Granular visibility: Records detailed information for each dropped packet, including the source IP address, destination port, and the specific network policy responsible for the decision.
- Faster troubleshooting: Allows administrators to quickly determine whether an OVN-Kubernetes access control list (ACL) denies traffic, eliminating guesswork during connectivity issues.
- Security compliance: Provides a verifiable audit trail that helps security teams confirm that default-deny policies are enforced correctly and detect potential lateral movement or unauthorized access attempts within the cluster.
- Policy refinement: Enables administrators to assess the impact of policies before enforcing them strictly in production by identifying legitimate traffic that can otherwise be blocked, helping to prevent unintended service disruptions.

Configuration

Instruct your cluster administrator to apply the following configuration:

1. Apply cluster configuration.
   If logging is not configured, the cluster administrator must patch the cluster network operator.

   Note: Consult your security team if they have a preferred audit log destination. Setting the destination to `libc` forwards the logs to the node's `systemd-journald`. Ensure `maxFileSize` is measured in bytes, not megabytes (for example, 50000000 is 50 MB).

   ```
   oc patch network.operator.openshift.io cluster --type=merge -p '{
     "spec": {
       "defaultNetwork": {
         "ovnKubernetesConfig": {
           "policyAuditConfig": {
             "destination": "libc",
             "maxFileSize": 50000000,
             "maxLogFiles": 5,
             "rateLimit": 20,
             "syslogFacility": "local0"
           }
         }
       }
     }
   }'
   ```

2. Find the logs.
   After you apply the configuration, query the audit log destination for deny alerts.

   If you configured the destination as `libc`, query the system journal on the nodes where the application pods are running to see the dropped traffic:

   ```
   oc debug node/<node-name> -- chroot /host journalctl | grep acl_log
   ```

Virtualization issues

This topic covers issues related to virtualization and their resolution.

GPU node power cycle issues

Problem statement

After a planned or unplanned power cycle of a GPU node, any virtual machines (VMs) that use a vGPU enter a `CrashLoopBackOff` state.

Resolution

1. Restart the `vgpu-device-config` pod in the `nvidia-gpu-operator` namespace.
2. Verify the available vGPU profiles by running the following command:

   ```
   oc get node <gpu_node_name> -o json | jq '.status.allocatable | with_entries(select(.key | startswith("nvidia.com/"))) | with_entries(select(.value != "0"))'
   ```

   Example output:

   ```
   {
     "nvidia.com/NVIDIA_RTX_Pro_6000_Blackwell_DC-48Q": "4"
   }
   ```

3. Start the VM.

Frequently asked questions

Commonly asked question and answers on deployment, storage, and backup of IBM Fusion.

- **Install**
  This section answers questions related to installation of IBM Fusion.

- **IBM Fusion Access for SAN**
  This section answers questions that are related to IBM Fusion Access for SAN.
- **Data Foundation Storage**
  This section answers questions that are related to storage in IBM Fusion.
- **Backup and Restore**
  This section answers questions related to backup and restore feature of IBM Fusion.
- **IBM Data Cataloging**
  This section answers questions that are related to IBM Data Cataloging in IBM Fusion.
- **Content-Aware Storage (CAS)**
  This section answers questions that are related to Content-Aware Storage in IBM Fusion.
- **Support and serviceability**
  This section answers questions related to serviceability.

Install

This section answers questions related to installation of IBM Fusion.

1. What are the supported deployments?
   Yes, you can install IBM Fusion on OpenShift® Container Platform that runs on On-premises VMware, On-premises Bare Metal, Linux on IBM Z, Microsoft Azure, Amazon Web Services, IBM Cloud®, and IBM Power Systems.

2. What are IBM Fusion services supported on the different deployment platforms for IBM Fusion?
   For more information the support matrix, see IBM Fusion services and platform support matrix matrix.

3. What are the resource requirements for fusion software?
   To know about resource requirements, see System requirements.

4. Does IBM Fusion support disconnected installation?
   Yes, it supports offline installation.

5. Is Deployment Guide available for IBM Fusion?
   The deployment guide is available in the following IBM Documentation location: Deploying IBM Fusion.

6. IBM Fusion can be deployed on UPI-installed OpenShift running on VMware? Does that imply that IBM Fusion cannot be deployed if OpenShift is installed using IPI for VMware?
   IBM Fusion assumes a bring-your-own OpenShift model. In other words, it assumes that OpenShift Container Platform is already installed. IBM Fusion does not dictate how it would or should have been installed.

7. Can IBM Fusion run on AWS ROSA and IBM Cloud ROKS? Can IBM Fusion run on public cloud (IBM Cloud, AWS) non-managed OpenShift environment?
   Yes, the IBM Fusion runs on AWS ROSA and IBM Cloud ROKS and also on public cloud (IBM Cloud, AWS) non-managed OpenShift environment.

IBM Fusion Access for SAN

This section answers questions that are related to IBM Fusion Access for SAN.

1. How do I use IBM Fusion Access for SAN with the Fusion Essentials license?
   IBM Fusion Access for SAN is supported with the Fusion Essentials license. It is not required to use Fusion Advanced or higher tiers to enable IBM Fusion Access for SAN.

2. How is storage capacity measured for IBM Fusion Access for SAN licensing?
   Licensing is based on usable managed SAN-attached storage capacity. It corresponds to the logical size of the volumes that are presented to IBM Fusion Access for SAN, regardless of how the underlying disks are laid out.

3. Does IBM Fusion Access for SAN distinguish between front-end and back-end storage capacity?
   No. There is no front-end or back-end distinction. The entitlement applies to the total amount of SAN-attached storage managed by IBM Fusion Access for SAN.

4. Are storage entitlements measured in TB or TiB?
   For licensing purposes, IBM defines 1 terabyte (TB) as 2^40 bytes. In practice, TB and TiB are treated as equivalent for IBM Fusion Access for SAN licensing.

5. How is backup capacity counted for IBM Fusion Access for SAN?
   Backup entitlement depends on the IBM Fusion edition:
   - Fusion Essentials: Backup is not included.
   - Fusion Advanced: Includes 8 TB of backup capacity per worker node.
   - Fusion Advanced Managed Server: Includes 12 TB of backup capacity per cluster.

   The Backup & Restore application calculates the backup usage based on the consumed backup storage.

6. Is the IBM Fusion operator required before installing IBM Fusion Access for SAN?
   No. The IBM Fusion Access for SAN operator is independent and does not require the IBM Fusion operator to be installed first.

7. Which container image registry is required for IBM Fusion Access for SAN?
   IBM Fusion Access for SAN requires access to a persistent local OpenShift® internal registry or a customer-provided internal registry (for example, Quay or JFrog). It is required for building and distributing GPFS kernel modules by using the Kernel Module Management (KMM) framework.
   This requirement applies to both connected and disconnected (air-gapped) environments.
   Run the following command to verify:

   ```
   $ oc get -o jsonpath='{.spec}' configs.imageregistry.operator.openshift.io/cluster | jq '. | (.managementState, .storage)'
   ```

The output must look similar to the following:

```
"Managed"
{
  "managementState": "Managed",
  "pvc": {
    "claim": "image-registry-storage" <The name of the PVC that is being used>
  }
}
```

8. Why does IBM Fusion Access for SAN fail with the error “Key was rejected by service” when loading the GPFS kernel module?
   This error typically occurs when Secure Boot is enabled on the worker nodes and the GPFS kernel module is:

   - Unsigned, or
   - Signed with a key that is not trusted by the system firmware.

   For more information to resolve this error, see https://examples.openshift.pub/storage/ibm-fusion-access-san/#if-secure-boot-is-enabled-create-and-roll-out-your-signing-key.

9. Is Secure Boot supported with IBM Fusion Access for SAN?
   Yes. Secure Boot is supported. However, additional configuration is required. The system firmware must explicitly trust the signing key that KMM uses. Without that trust, the kernel module load fails. For more information, see IBM Fusion Access for SAN using Secure Boot.

10. What are my options if Secure Boot is enabled and the module fails to load?
    You have two supported options:

    a. Manually trust the signing key used by KMM in the system firmware (recommended if Secure Boot is required).
    b. Disable Secure Boot on the worker nodes if the customer environment does not require it.

    If the wrong key is used during initial installation, a full cleanup and redeployment of IBM Fusion Access for SAN might be required.

11. During an IBM Fusion Access for SAN upgrade, are virtual machines automatically drained or stopped?
    IBM Fusion Access for SAN upgrades rely on underlying OpenShift and Storage Scale (CNSA) upgrade behavior. In general:

    - Storage Scale components are upgraded after other workloads are drained.
    - OpenShift Virtualization primarily controls the VM behavior during upgrades.

    Red Hat provides the authoritative behavior for VM lifecycle during upgrades.

12. Can I control the order in which nodes are upgraded?
    You cannot explicitly define an exact node-by-node upgrade order. However, you can:

    - Control `maxUnavailable` settings
    - Pause and resume upgrade pools
    - Limit parallel upgrades

    These mechanisms allow manual, controlled upgrades, including effectively upgrading one node at a time.

13. Is manually pausing upgrade pools to control node upgrades supported?
    Yes. Storage Scale Container Native (CNSA) supports this method. While it is not explicitly tested as a IBM Fusion Access for SAN-specific workflow, it is expected to work when CNSA supports it. Validate this approach in non-production environments.

14. Is IBM Fusion Access for SAN supported with IBM Fusion HCI?
    No. IBM Fusion Access for SAN is not supported with IBM Fusion HCI deployments.

15. Can IBM Fusion Access for SAN be used in a cluster running both containers and virtual machines?
    Yes. IBM Fusion Access for SAN can provide storage to:

    - Container workloads
    - OpenShift Virtualization virtual machines
    - Nested or test OpenShift clusters running as VMs

16. Are Hosted Control Plane (HCP) clusters supported with IBM Fusion Access for SAN?
    No. Hosted Control Plane (HCP) clusters are not supported. IBM Fusion Access for SAN cannot be used as storage for hosted clusters.
    The base OpenShift cluster might use IBM Fusion Access for SAN, but hosted clusters cannot.

17. Is IBM Fusion Access for SAN supported on IBM Z (s390x) and IBM LinuxONE platforms?
    No. IBM Fusion Access for SAN is not supported on IBM Z (s390x) and IBM LinuxONE platforms.

18. Can external worker nodes use IBM Fusion Access for SAN in a mixed environment?
    In scenarios where external worker nodes are present and connected to external SAN storage, those workers might use IBM Fusion Access for SAN, provided the deployment aligns with IBM Fusion SDS support boundaries.

19. After upgrading IBM Fusion Access for SAN, why does the filesystem no longer appear in the web console even though storage is still working?
    Starting with IBM Fusion Access for SAN v1.1, the UI relies on the **`FilesystemClaim`** (FSC) resource. If an existing filesystem is not created by using the IBM Fusion Access UI or FSC workflow, the UI cannot display it. Storage continues to operate because the IBM Spectrum Scale operator still manages the filesystem, but the storage no longer appears in the IBM Fusion Access UI.

20. Can I manually create a FilesystemClaim for an existing filesystem created outside the IBM Fusion Access UI?
    No. IBM Fusion Access for SAN reconciles only filesystems that it creates using its opinionated UI or **`FilesystemClaim`** workflow. Existing filesystems that are created manually or by automation cannot be retroactively attached to a **`FilesystemClaim`**.

21. Does upgrading from IBM Fusion Access for SAN v1.0 to v1.1 automatically migrate existing filesystems?
    Yes, only if the filesystem is originally created using the IBM Fusion Access for SAN v1.0 UI. In that case, the upgrade automatically creates the corresponding **`FilesystemClaim`** and the UI continues to work as expected. Filesystems created manually are not migrated.

22. Why is creating filesystems directly through YAML no longer supported in newer versions?
    IBM Fusion Access for SAN is deliberately opinionated. From v1.1 onward, the product enforces a single management path using **`FilesystemClaim`** to simplify lifecycle management. Hybrid workflows (partial UI usage and partial manual YAML management) are not supported.

23. What is the relationship between the IBM Fusion Access for SAN operator and the IBM Spectrum Scale operator?
    The IBM Fusion Access for SAN operator:

    - Installs and configures the IBM Spectrum Scale (CNSA) operator
    - Provides a UI and high-level abstractions such as **`FilesystemClaim`**

    The IBM Spectrum Scale operator performs the actual filesystem, disk, and GPFS management. IBM Fusion Access adds automation and an abstraction layer, not a separate storage implementation.

24. Why do I see multiple default **`StorageClasses`** after upgrading to IBM Fusion Access for SAN v1.1?
    This is a known behavior in v1.1. Multiple **`StorageClasses`** might be marked as default after an upgrade. It does not break functionality and does not prevent users from selecting a different **`StorageClass`**.

25. If my internal image registry data is lost, how can I force IBM Fusion Access for SAN to rebuild GPFS kernel modules?
    Delete and recreate the **`Module`** custom resource in the Kernel Module Management (KMM) operator. This triggers a rebuild and republish of the kernel module into the internal registry. After the build completes, the Scale pods reconcile automatically or after a pod restart.

26. Can IBM Fusion Access for SAN coexist with other products that install IBM Spectrum Scale (for example, GDP remote mount service)?
    No. Multiple GPFS/Scale versions cannot coexist in the same OpenShift cluster. If another product installs a different Scale version, kernel module conflicts occur. Uninstall the other Scale-based service before installing IBM Fusion Access for SAN.

27. Is a GPFS kernel module load failure always caused by an unsupported OpenShift version?
    No. Most observed failures are related to KMM, Secure Boot, or conflicting GPFS installations, not OpenShift version incompatibility. OpenShift upgrades must be only considered after eliminating these causes.

28. Can IBM Fusion Access for SAN support mixed worker nodes (bare metal with FC and VMs with iSCSI) in the same cluster?
    No. IBM Fusion Access for SAN expects all participating nodes to access the same disks using identical access methods. Mixing FC-attached bare-metal nodes and iSCSI-attached VM nodes to the same filesystem is not supported.

29. Can IBM Fusion Access for SAN be deployed only on a subset of worker nodes?
    Yes. IBM Fusion Access for SAN is deployed only on nodes where SAN LUNs are visible. Workloads must be scheduled by using labels or node selectors to ensure that they run on those nodes. However, client-only nodes (no disk access) are not supported, which means that nodes without LUNs cannot run persistent workloads or access the PVs.

30. Is disaster recovery (DR) between two IBM Fusion Access for SAN clusters supported today?
    No, cross-cluster DR is not supported today. DR capabilities leveraging storage replication and Kubernetes-level orchestration are planned on the roadmap. Customers must currently rely on underlying storage replication and manual recovery processes.

31. Why does **`FileSystemClaim`** fail with "disk has Spectrum Scale filesystem data structures on it"?
    The disk might contain GPFS metadata from a previous deployment. IBM Fusion Access for SAN prevents reusing such disks to avoid data corruption.

32. How can I reuse disks that previously contained a Spectrum Scale filesystem?
    You must fully clean the disks before reuse. This can include:
    - Removing partitions
    - Clearing multipath mappings
    - Zeroing disk headers (for example, **`dd if=/dev/zero`**)
    - Removing leftover host files under /var/mmfs and /var/adm/ras on the worker nodes

    After cleanup, recreate the **`FileSystemClaim`**.

33. Can I create the **`FileSystemClaim`** before local volume discovery results are available?
    No. The supported workflow requires that **`LocalVolumeDiscoveryResults`** exist before the **`FileSystemClaim`** (FSC) is created.
    If the FSC is created before discovery completes, validation fails and the FSC does not automatically recover, even after discovery results appear. The FSC must be deleted and recreated once discovery is complete.

34. Does **`FileSystemClaim`** automatically retry when discovery results appear later?
    No. In this scenario, automatic retries are not triggered once the initial validation fails.
    Although the controller retries reconciliation in other cases. FSC validation failures that occur early in the reconciliation loop do not trigger retries unless:

    - The FSC **`spec`** is updated, or
    - The controller is restarted.

35. What is the recommended deployment order when automating the installation of IBM Fusion Access for SAN?
    When automating deployment, IBM recommends the following sequence:
    a. Deploy operators
    b. Create the **`FusionAccess`** custom resource
    c. Wait for local volume discovery results
    d. Create the **`FileSystemClaim`**

    This is required because IBM Fusion Access is operator-driven but also installs and manages another operator (IBM Spectrum Scale).

36. Why do Filesystem Claims disappear after an OpenShift upgrade and power outage?
    Filesystem Claims might disappear if the environment enters an unsupported or inconsistent state, such as:
    - Partial recovery following a system-wide outage
    - Kernel module or registry failures leading to mismatched Spectrum Scale versions

    In such cases, the UI can show a blank state even though underlying storage objects still exist. If the issue persists, open a support case to help you stabilize your system.

37. Can Filesystem Claims be recreated manually after they disappear?
    No. Filesystem Claims cannot be reconstructed from existing Filesystem or Local Disk objects.
    IBM Fusion Access for SAN only manages and reconciles resources that it originally created. If Filesystem Claims are missing and the environment is inconsistent, recovery might require redeployment of IBM Fusion Access for SAN. If the issue persists, open a support case to help you stabilize your system.

38. Is it supported to recover Filesystem Claims after disk-level or metadata rollback?
    No. Recovery from partial disk restores or metadata rollbacks that result in mixed Spectrum Scale versions or missing IBM Fusion artifacts is not supported.
    If underlying disk metadata or Scale versions are inconsistent, the recommended approach is to rebuild the environment cleanly.

39. How can I verify which Spectrum Scale version is running?
    You can verify the active Spectrum Scale version by:

    - Inspecting the Cluster CR
    - Running commands inside a Spectrum Scale core pod, such as:

        - `mmdiag --version`
        - `rpm -qi gpfs.base`
        - `mmlsfs all <to check filesystem format levels>`

40. Is IBM Fusion Access for SAN affected by direct I/O performance issues that are seen in containerized Scale deployments?
    Earlier containerized deployments might experience Direct I/O performance degradation due to permission constraints introduced in RHEL 9, where unprivileged containers can encounter EPERM errors when accessing block devices.
    This behavior is addressed in IBM Storage Scale v5.2.3, which avoids dropping out of Direct I/O mode under these conditions.

41. Does IBM Fusion Access for SAN require Kubernetes device plug-ins (such as power-device-plugin) for SAN performance?
    No. IBM Fusion Access for SAN does not require device plug-ins for SAN access or Direct I/O.
    SAN LUNs are mapped directly to worker nodes at the operating system level, and IBM Storage Scale builds on top of those devices using a shared-disk architecture. Device plug-ins are typically relevant for specific virtualized or accelerator-focused use cases, not standard IBM Fusion Access deployments.

42. Does IBM Fusion Access for SAN support using IBM Spectrum Scale QoS (mmqos) commands to control I/O?
    No. IBM Fusion Access for SAN does not support the use of CNSA QoS (mmqos) for I/O throttling or prioritization.

43. Can file systems created by IBM Fusion Access for SAN be encrypted?
    IBM Fusion Access for SAN does not support CNSA encryption for IBM Fusion Access–managed file systems.
    However, SAN-based storage can provide at-rest encryption at the LUN level, which can be used instead. This is the recommended method if encryption is required.

44. Is IBM Fusion Access for SAN dependent on a full IBM Fusion SDS deployment, or can it run as a standalone operator?
    Yes. IBM Fusion Access for SAN can run as a standalone operator on OpenShift.

45. Is IBM Fusion Access for SAN supported with OpenShift running on PowerVM?
    No. IBM Fusion Access for SAN is not supported in this environment.

Data Foundation Storage

This section answers questions that are related to storage in IBM Fusion.

1. Does the Data Foundation support Object Storage?
   Yes, the Data Foundation support Object Storage.

2. What are the supported deployments?
   All IBM Fusion services are supported on the Bare Metal platform of IBM Fusion. For supported deployments, see IBM Fusion services and platform support matrix.

3. What is the supported Data Foundation version?
   The Data Foundation version is the same as the OpenShift® Container Platform version, for example, on OpenShift Container Platform 4.16, install Data Foundation 4.16.
   For a complete list of supported versions, see Support matrix for IBM Fusion (SDS) versions.

4. What is the difference between Fusion Data Foundation and Red Hat® OpenShift Data Foundation?
   Fusion Data Foundation and Red Hat OpenShift Data Foundation provide similar functionality for provisioning container-native storage in an OpenShift cluster. IBM allows Red Hat to resell support for ODF through a subscription model, with some minor differences in capability. For example, Provider mode is not available in ODF. Red Hat offers ODF in two editions: ODF Essentials and ODF Advanced, with ODF Advanced including Disaster Recovery (DR) capabilities. Currently, neither IBM Fusion nor IBM Cloud Paks include a subscription to Red Hat ODF. The FDF image is sourced from the IBM image repository, while the ODF image is sourced from the Red Hat image repository.

5. What levels of encryption are supported?
   Both cluster-wide encryption and storageclass encryption are supported.

6. How to handle and support key management?
   Support is available for external Key Management System, either HashiCorp Vault or Thales CipherTrust Manager.

7. Does Fusion Data Foundation support encryption of data in flight?
   Yes, it utilizes Transport Layer Security (TLS).

Backup and Restore

This section answers questions related to backup and restore feature of IBM Fusion.

1. Can I install the Backup & restore as a day 2 operation?
   Yes, from the Services user interface page of IBM Fusion.

2. How to verify whether Velero was installed and what gets backed up behind the scenes?
   Velero runs as a pod in the **`ibm-backup-restore ns`** namespace. If you are in the OpenShift® Container Platform console, look at the pods in the **`ibm-backup-restore ns`** namespace or run the following **`oc`** command:

```
oc get pods -n ibm-backup-restore ns
```

3. What are the supported backup storage locations?
   You can use IBM Fusion to create both local backups as well as copy data to an external, even offsite, S3 compliant object store.

4. Does IBM Fusion backs up application data consistently across all the Persistent Volumes (PV)? Which applications are supported in an application consistent backup mode within IBM Fusion?
   Yes, it backs up application data consistently across all the PVs. There are two ways to achieve consistency. One is to quiesce the application. That can be done for virtually any application, but it does imply the willingness to accept some downtime. Another way is crash consistent backup. Not all applications may be able to restore themselves using a crash consistent backup. It depends on the application and how it is designed. For more information about data consistency support in IBM Fusion, see Achieving data consistency for containers.

5. Will IBM Fusion data protection support ODF remote mount?
   Yes, ODF supports a remote mount. For more information, see Deploying OpenShift Data Foundation in external mode in *Red Hat OpenShift Data Foundation* documentation.

IBM Data Cataloging

This section answers questions that are related to IBM Data Cataloging in IBM Fusion.

1. What is IBM Data Cataloging?
   IBM Data Cataloging is an IBM Fusion service that provides the capability of analyze the metadata of customer's data sources. It provides rapid automated data discovery and robust metadata capture, curation an enrichment.

2. How can you perform that data analysis and metadata capture?
   In first place, you need to create connections to the wanted data source. After that, you need to select the connection and run a scan over it. You see different statuses when running your scan (as scanning or indexing, for example). After terminated, you see your data that is indexed in the IBM Data Cataloging database.

3. What other actions can you do with your data?
   You can do more complex management with your data by adding tag management or running specific policies.

4. What kind of data sources are supported?
   IBM Data Cataloging supports connection to several kinds of data sources, such as IBM Storage Scale, NFSv4, S3, Cloud Object Storage, IBM Storage Protect, and Server Message Block.

5. Can we use IBM Storage Scale as storage provider for IBM Data Cataloging?
   Yes, IBM Storage Scale can be used as storage provider to install IBM Data Cataloging. You need to have a Scale cluster to be set as remote storage provider, and use the Global Data Platform service available in IBM Fusion to connect your Fusion cluster with the remote Scale.

6. Can we use Red Hat® OpenShift® Data Foundation as storage provider for IBM Data Cataloging?
   Yes, Red Hat OpenShift Data Foundation can be used as storage provider for IBM Data Cataloging. To use it, install the IBM Data Cataloging service available in IBM Fusion. After installed, you can install IBM Data Cataloging. When installing IBM Data Cataloging, be sure that you select the `ocs-storagecluster-cephfs` as storage class.

7. Can we use any storage class to install IBM Data Cataloging?
   IBM Data Cataloging was designed to be storage-agnostic, which means that you can use any storage provider. Only ensure that the storage class selected meets with the requirements that are specified in the prerequisites section.

Content-Aware Storage (CAS)

This section answers questions that are related to Content-Aware Storage in IBM Fusion.

1. What is Content-Aware Storage (CAS) in IBM Fusion Systems?
   CAS is a storage service that helps unlock the value of unstructured data through AI technologies. It enables the access to this value in new data-driven solutions, without the complexities of implementing and managing large AI solutions.

2. What is the primary objective of the CAS architecture?
   The CAS architecture is designed to enable seamless interaction between large language models (LLMs) and vast volumes of unstructured data, enhancing insight generation and suggestion capabilities.

3. How does CAS process unstructured data?
   CAS automatically processes files that are stored on a remote IBM Storage Scale cluster. It extracts parsed content and normalizes metadata, which is then stored along with vector embeddings in a managed vector database.

4. What type of storage platforms does CAS support?
   CAS supports IBM Storage Scale software-based clusters, and Storage Scale Server 3500 and 6000 platforms. It uses remote mounting by using the IBM Fusion Global Data Platform service.

5. How does CAS detect and ingest modified data?
   CAS uses IBM Storage Scale watch folders to detect changes (additions, deletions, or updates) in the file system. Only the modified content is ingested, optimizing performance and resource usage.

6. Can CAS ingest data from external S3 sources?
   Yes, CAS supports ingest processing of files that are stored in external S3 sources by using Active File Management (AFM). It automatically scans AFM-S3 filesets to identify data for ingestion.

7. What type of document processing services does CAS support?
   CAS integrates with IBM's Docling Multimodal services and NVIDIA NeMo Retriever Extraction microservices for document parsing and content extraction.

8. How does CAS perform search operations?
   CAS provides a search API that supports semantic, keyword, and hybrid search methods. It aggregates results and optionally uses NVIDIA's reranking for enhanced accuracy.

9. How do customer applications interact with CAS?
   Applications such as chatbots can send natural language queries to the CAS search API. CAS converts the query into a vector representation and performs a search against its vector database, returning optimized results.

10. What is the role of NVIDIA reranking service in CAS search?
    In the NVIDIA multimodal pipeline, CAS uses an aggregator component to refine the top 100 search results. These results are further optimized by using keyword-based and optional reranking before they are returned to the API.

11. Does CAS support Retrieval-Augmented Generation (RAG) workflows?
    Yes, CAS search results can be combined with user prompts and sent to an LLM to generate responses, enabling RAG-based client applications.

12. What types of data does CAS support for semantic extraction?
    CAS focuses on unstructured text data. It uses natural language processing (NLP) and AI technologies to extract semantic meaning from documents.

13. Which IBM Fusion version is required to install CAS?
    CAS requires the newest IBM Fusion version. Installation must be performed by using the default **`ibm-spectrum-fusion-ns`** namespace. Custom namespaces are not supported for CAS installation.

14. What are the hardware requirements for CAS?
    To run CAS on three OpenShift® compute nodes, the following resources are required:
    - 3 × NVIDIA GPUs
    - One other GPU for optional reranker service
    - 166 vCPUs (83 physical cores)
    - 1024 GB memory
    - 1 TB IBM Storage Scale

15. What is the recommended IBM Storage Scale configuration?
    The most common configuration includes:
    - 48 NVMe drives
    - 30 TB capacity (Enterprise Storage Server 6000)
    - 4 CX-7 network adapters
    - 1.5 TB memory

16. What software versions are required for IBM Storage Scale?
    Ensure that IBM Storage Scale version 5.2.3.1 or later is installed. Remote file systems must be configured and available for CAS.

17. How do you verify the availability of a storage class for CAS?
    In the OpenShift console, go to Storage > Storage Class and confirm that the storage class (for example, **`ibm-spectrum-fusion`**) created from the IBM Storage Scale remote file system is available and marked as default.

18. What are the prerequisites for enabling GPU workloads in OpenShift?
    Install the following components:
    - Red Hat Node Feature Discovery Operator
    - NVIDIA GPU Operator (compatible version)

19. What is NVIDIA NIM and why is it required?
    NVIDIA NIM is a set of optimized microservices for deploying generative AI models. CAS uses NVIDIA NeMo Retriever for document processing. Provide NVIDIA NGC license keys to pull components from the NVIDIA NGC registry.

20. Is the NVIDIA text reranker mandatory?
    No, the NVIDIA text reranker is optional and requires an additional supported GPU. It enhances search result accuracy by using advanced scoring.

21. What type of proxy URLs must be allowlisted for CAS installation?
    Add the following URLs to your proxy allowlist:
    - **`apigee.googleapis.com`**
    - **`apigeeconnect.googleapis.com`**
    - **`binaryauthorization.googleapis.com`** (optional)
    - **`gcr.io`**
    - **`raw.githubusercontent.com`**

22. How do you configure CAS to use the NVIDIA reranker service?
    Add the following flags to the **`cas-config`** ConfigMap:
    - **`NVMM_NEMO_RANKER`**
    - **`NVMM_NEMO_RANKER_SERVICE`**

23. What user permissions are required for remote file system setup?
    The Global Data Platform user must be part of the **`StorageAdmin`** group to enable watch creation. Membership in **`CsiAdmin`** or **`ContainerOperator`** alone is insufficient.

24. How do you initiate the installation of CAS?
    You can start the installation from the IBM Fusion user interface by navigating to the Services page and selecting the Content-Aware Storage tile.

25. What components are installed with CAS?
    The CAS installation includes Kafka, the CAS operator, and CNPG (Cloud Native PostgreSQL).

26. What prerequisites must be met before installing CAS?

Ensure the following items:

- All prerequisites that are listed in the Planning and prerequisites section are fulfilled.
- A default storage class (for example, `ibm-spectrum-fusion`) is available and marked as default.
- All IBM Storage Scale projects and pods are running.

27. What can you do if the installation pre-check raises a warning?
    Resolve the warning at the OpenShift level before proceeding. The pre-check ensures that the environment is ready for CAS deployment.

28. How can you monitor the CAS installation progress?
    After initiating installation, a notification appears on the Services page. You can track progress under Services > Installed. After completed, the CAS service status shows as Healthy.

29. How do you verify that CAS is installed successfully?
    You can verify installation from:
    - **IBM Fusion UI**: Check service version and health status. Use the ellipsis menu to download logs and view documentation.
    - **OpenShift Console**: Go to Home > Projects and search for a namespace that contains `ibm-cas`. Then check Operators > Installed Operators for CAS availability and service version.

30. How do you configure CAS after installation?
    Set up a Scale user to enable watch creation, configure domains and data sources, and map domains to data sources. For more information, see Configuring Content-Aware Storage (CAS).

31. Can you download installation logs?
    Yes. From the ellipsis menu in the CAS service window, you can download logs and view related documentation. A success notification confirms log collection.

32. What happens if the CAS service status is not healthy after installation?
    Check the logs for errors, verify that all prerequisites are met, and ensure that all required services (Kafka, CNPG, CAS operator) are running. Revalidate the environment and retry installation if needed.

Support and serviceability

This section answers questions related to serviceability.

1. How can I display events from the command line?
   The following OC command lists a summary of each IBM Fusion event currently in the event queue:

   ```
   oc get events --field-selector reason=ISFEventManager
   ```

   The following OC command lists the contents of each IBM Fusion event currently in the event queue:

   ```
   oc get events --field-selector reason=ISFEventManager -o yaml
   ```

2. How long do events stay in the event queue?
   - INFO events stay in the queue 3 hours just like regular OCP events
   - WARNING events stay in the event queue 7 days
   - CRITICAL events stay in the event queue 14 days

3. How can I delete events?
   The events are deleted as per default rules specified in question 2 of this section.

4. How can I mark a Call Home ticket as closed?
   There is no automatic update of the ISF events from the Call Home server. After a ticket gets closed, you must go to the ISF Events UI, select the event that opened the ticket, and then in the events context menu, choose "Mark as fixed".

5. Is the IBM Call Home feature mandatory?
   Call Home is an optional feature. Customers can disable the feature if they have concerns that prevent them from giving IBM Fusion access to the internet.

6. What collection sets are defined?
   There are 2 main config maps that define what logs will be collected. isf-logs is a list of individual requests

   ```
   logcollector-dep:
         type: k8s-resource
         group: apps
         version: v1
         kind: deployments
         namespace: ibm-spectrum-fusion-ns
         name: logcollector
         description: Log Collector log
   ```

   This entry in `isf-logs` indicates that when the `logcollector-dep` request is made, Log Collector will gather all information about the `logcollector` deployment in the `ibm-spectrum-fusion-ns` namespace (including its pods and pod logs). The second config map is `isf-collection-sets`, which can group requests from `isf-logs`, other entries in `isf-collection-sets`, and individual requests.

   ```
   compute:
         type: collection-set
         collections:
           must-gather:
             type: list-entry
             source-list: isf-collection-sets
             list-entry: must-gather
           ibm-spectrum-fusion-namespace:
             type: list-entry
   ```

```
        source-list: isf-logs
        list-entry: ibm-spectrum-fusion-namespace
```

This entry in `isf-collection-sets` indicates that when the compute collection set is requested, Log Collector will gather the must-gather collection set (defined in `isf-collection-sets`) and `ibm-spectrum-fusion-namespace` (defined in `isf-logs`).

7. What files are included in each collection set shown on the Log Collector UI?
   The specific contents will change as more functions are added to the product, but for now a good summary is in the "Collection Contents" section of this blog post: Viewing and Examining Logs in IBM Spectrum Fusion

8. How long should it take for logs to be collected from the UI?
   All collection sets must finish within 10 minutes, except for the "System health check", which takes 2 to 3 hours. Sometimes, it can take up to a maximum of 6 hours to run.

9. How long will log collections remain on the server?
   Logs will be automatically deleted after 24 hours. Logs can be intentionally deleted from the UI page.

10. If my log collection status is partial, does it indicate a log collection failure?
    Partial state of log collection does not mean that the process failed during log collection. It indicates the failure of at least one item in the collection, but no impact to other logs in the collection.

11. What filter options are available for the IBM Fusion Events page?
    See the "Event UI" section of this blog post: Viewing and interpreting events in IBM Spectrum Fusion

12. What services are sold with IBM Fusion?
    Today, support will be provided via IBM Expert Care contracts. Add on services will be developed later.

13. What is IBM Storage Expert Care?
    Storage Expert Care is a simplified method of selecting services and support for storage systems at the time of purchase. Storage Expert Care is designed to simplify and standardize the support approach on IBM Fusion, FlashSystem 5200, 7200 & 9200 with pricing calculated as a percentage of the net system (HW & SW). Clients may choose the support level and duration, to align with their business needs.

14. Must clients purchase Storage Expert Care?
    A client has the option of 1) buying the hardware with the Warranty alone, or 2) choosing from Storage Expert Care Basic or Premium on IBM Fusion. The standard Warranty is: 1 Year of Warranty services, 9x5xNBD, Parts-only, IOL Limited and 90-day SWMA. IBM does not offer separate HW or SW maintenance for these products except for Storage Expert Care.

15. Which Storage Expert Care tiers are available?
    Basic and Premium for IBM Fusion was announced on 8/10/21.Support terms are available for1, 2, 3, 4, and 5 years.

16. What are the components of the Premium tier?
    Premium tier includes IBM Software Maintenance (SWMA), Technical Account Manager (TAM), 30 Min. Enhanced Response (Sev1 and Sev2), and Predictive Support.

17. Is remote support available for upgrade?
    IBM offers twice a year updates and they can be applied remotely. However, the client must enable remote access to IBM Fusion. Support is not yet available for IBMers to remotely access client systems on-demand and without client interaction.

18. Where to find more information about Remote support?
    For more information about remote support, see Chapter 16 of our internal service guide- http://worklodepid1.fyre.ibm.com:8081/spectrum_fusion/9155_spectrum_fusion_24_service_book.pdf.

19. Will clients be able to purchase Storage Expert Care only if they selected Warranty at initial purchase?
    Storage Expert Care will be available for separate purchase in TSS Systems (CONGA) in 1Q22. IBM highly recommends selling Storage Expert Care together with the system at time of purchase to ensure the client's systems are protected beyond the standard Warranty. Standard warranty provides 90-day SWMA and 9x5xNBD, Parts-only, IOL Limited.

20. What services are sold with IBM Fusion?
    Today, support will be provided via IBM Expert Care contracts. Add on services will be developed later.

21. When will Renewals become available in TSS Systems (ISAT/CONGA)?
    Renewals will be available in TSS Systems (CONGA) in 1Q22. Prices will be based on the original services price and discount.

22. Will TSS be able to quote additional years than what is available in the configurator (years 1-5)?
    Yes, clients can request special bid pricing for years 6 and 7 of Storage Expert Care.

23. Which date will IBM consider as a start date for Expert Care after the order?
    Storage Expert Care starts at the same time as Warranty start. IBM has published rules for Warranty start for Customer Setup machines (CSU) and IBM Installed (IBI) machines.

24. What is Predictive Support?
    IBM provides predictive alerts for performance, space capacity, and other system problems. If there is a significant need for immediate action to be taken to avoid or prevent an incident, an action plan will be discussed with client.

25. How will customers get support for IBM Fusion? Will they need to manage multiple support contacts or will there be a single point of contact to receive support?
    There will be a single point of contact for support. The customer will open support tickets against IBM Fusion to receive support for both hardware and software. In some instances, the customer may be required to open support tickets with Red Hat to obtain support for OpenShift® Container Platform. IBM and Red Hat will collaborate to make the support experience as unified as possible.

26. Is Lab service team involved to help the client? If so, is it charged service or included in Expert Care premium?
    After IBM Fusion is installed and ready for customer usage, customers can then begin creation and deployment of containers. Customers will also be able to monitor the system containers/workloads, system resource utilization and alerts. IBM Fusion will support Premium Expert Care and will include the installation services.

27. Is Lab Services required to implement IBM Fusion in the Customer location?

IBM Fusion does not require lab services. IBM Fusion is physically installed into the customer datacenter by an IBM SSR. The SSR will inspect the hardware and connect power and network cables. The SSR will additionally perform necessary configuration steps to integrate the appliance in the customer's network.

Accessibility features for IBM Fusion

Accessibility features assist users who have a disability, such as restricted mobility or limited vision, to use information technology content or products successfully.

Accessibility features

IBM Fusion includes the following major accessibility features:

- Keyboard-only operation
- Operations that use a screen reader
- Interfaces that are commonly used by screen readers
- Keys that are discernible by touch but do not activate just by touching them
- Industry-standard devices for ports and connectors
- The attachment of alternative input and output devices

IBM® Documentation, and its related publications, are accessibility-enabled. The IBM Fusion online product documentation in IBM Documentation (IBM Docs) is enabled for accessibility. The accessibility features are described in IBM Documentation at Accessibility.

IBM Fusion uses the recent W3C Standard, WAI-ARIA 1.0 (www.w3.org/TR/wai-aria/), to ensure compliance with US Section 508 (www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards) and Web Content Accessibility Guidelines (WCAG) 2.0 (www.w3.org/TR/WCAG20/). To take advantage of accessibility features, use the recent release of your screen reader and the recent web browser that is supported by IBM Fusion.

Keyboard navigation

This product uses standard Microsoft Windows navigation keys and the following keyboard shortcuts.

Table 1. Keyboard shortcuts in IBM Fusion

Action	Shortcut for Internet Explorer	Shortcut for Firefox
Move to and around the navigation menu	Tab and up/down/left/right	Tab and up/down/left/right
Exit the navigation menu or close the dialog	Esc	Esc
Move to the Contents View frame	Enter	Enter

Interface information

The IBM Fusion user interfaces do not have content that flashes 2 - 55 times per second.

The IBM Fusion web user interfaces rely on cascading stylesheets to render content properly and to provide a usable experience. The application provides an equivalent way for low-vision users to use a user's system display settings, including high-contrast mode. You can control font size by using the device or web browser settings.

The IBM Fusion web user interface includes WAI-ARIA navigational landmarks that you can use to quickly go to functional areas in the application.

Exception and work-around

The following pages do not provide keyboard-only operation and a screen reader:

- Deploy Distribute UI
- Monitoring UI of deployment manage inlet
- Pattern builder UI

Related accessibility information

In addition to standard IBM help desk and support websites, IBM established a TTY telephone service for use by deaf or hard of hearing customers to access sales and support services:

TTY service 800-IBM-3383 (800-426-3383) (within North America)

IBM and accessibility

For more information about the commitment that IBM has to accessibility, see the IBM Human Ability and Accessibility Center.

IBM Fusion considerations for GDPR readiness

IBM Fusion considerations for GDPR readiness cover data classification, lifecycle management, and user account controls for protecting personal data.

Notice

This document is intended to help you in your preparations for GDPR readiness. It provides information about features of IBM Fusion that you can configure, and aspects of the product's use, that you must consider to help your organization with GDPR readiness. This information is not an exhaustive list, due to the many ways that clients can choose and configure features, and the large variety of ways that the product can be used in itself and with third-party applications and systems.

Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that might affect the clients' business and any actions the clients might need to take to comply with such laws and regulations.

The products, services, and other capabilities that are described here are not suitable for all client situations and might have restricted availability. IBM does not provide legal, accounting, or auditing advice or represent or warrant that its services or products ensure that clients are in compliance with any law or regulation.

Table of Contents

1. GDPR
2. Product configuration - considerations for GDPR readiness
3. Data life cycle
4. Data collection
5. Data storage
6. Data access
7. Data processing
8. Data deletion
9. Data monitoring
10. Responding to data subject rights

GDPR

General Data Protection Regulation (GDPR) is adopted by the European Union ("EU") and applies from 25 May 2018.

Why is GDPR important?
GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

Read more about GDPR

- EU GDPR Information Portal
- ibm.com/GDPR website

Product configuration – considerations for GDPR readiness

The following sections provide considerations for configuring IBM Fusion to help your organization with GDPR readiness.

Data life cycle

User accounts
The IBM Fusion system administrator or security administrator creates a user by providing a user ID, email address, full name, and password to grant the user access to the system. This personal data is stored in the database on the client's hardware and can be fully managed by the system administrator or security administrator and edited by the user.

Information on managing users is documented in IBM Documentation (IBM Docs) for IBM Fusion.

System logs
Personal data, including IP addresses, session IDs, user IDs, web-page URLs, and cookie names, might exist within operating system and application logs. The data within these logs is captured automatically as part of the offering and is beyond the control of the client. The logs are retained on disk when sufficient space is available. As more space is needed, older log files are removed. The log files might not be modified or deleted by the client.

The purpose of the system log files is for use during troubleshooting situations. As needed, the log files might be collected and downloaded from the offering for transfer to IBM Support. The log files are not included in the system backups and are therefore constrained to the node unless involved while logs are collected for troubleshooting activity.

Information on system logs is documented in IBM Documentation for IBM Fusion.

Personal data used for online contact with IBM
IBM Fusion clients can submit online comments/feedback/requests to contact IBM about IBM Fusion subjects in various ways, primarily:

- Public comments area on pages in the IBM Fusion community.
- Public comments area on pages of IBM Fusion documentation in IBM Documentation.
- Public comments in the IBM Fusion space of dWAnswers
- Feedback forms in the IBM Fusion community

Typically, only the client name and email address are used, to enable personal replies for the subject of the contact, and the use of personal data conforms to the IBM Online Privacy Statement.

Data collection

For more information, see Data life cycle.

Data storage

Personal data is contained within backups of the offerings. Such personal data includes the personal data that is associated with user accounts that are stored within the database. The IBM Documentation provides information pertaining to creating the backups within the IBM Fusion offering.

The backup feature enables the client to transfer the backup archives to an external location. However, management of any external backup archives is beyond the scope of the offering. The client must implement a set of established 'best practices' for managing and securing such backup files. Information on managing backups is documented in IBM Documentation for IBM Fusion.

Data access

General security measures (for example, disk encryption, physical and remote access) either directly implemented by the offering or suggested actions for the client's preparedness to deploy the offering are documented in IBM Documentation.

For user account data, read or write access can be given to specific users.

Data processing

General security measures are directly implemented by the offering.

Data deletion

Personal data associated with user accounts (as described in Data life cycle) can be fully managed by the system administrator or security administrator, including deletion. Users are not authorized to delete the personal data associated with the accounts. Information on managing users is documented in IBM Documentation for IBM Fusion.

Personal data, including IP addresses, session IDs, and user IDs, might exist within operating system and application logs. The log files might not be modified or deleted by the client. The logs are retained on disk when sufficient space is available. As more space is needed, older log files are removed. The log files might not be modified or deleted by the client. Information on system logs is documented in IBM Documentation for IBM Fusion.

Data monitoring

IBM Fusion does not monitor operating system or application logs, which are collected by the system and remain on the node as space permits. When needed for troubleshooting, logs might be downloaded from the console. Typically, such files remain local to the offering and cannot be managed or altered by users or administrators. Administrators might be able to review some log files (for troubleshooting purposes and without context of any personal data that is contained within) from the offering console. For more complex troubleshooting situations, such logs might be collected and downloaded from the offering for transmission to IBM Support.

Responding to data subject rights

IBM Fusion meets the following data subject rights: right to access, modify, forgotten, and portability.

Glossary

Glossary terms and definitions cover key storage, networking, Kubernetes, and AI concepts used across storage and data management products.

This glossary provides terms and definitions for the #REPLACE_ME_PRODUCTS# software and products.

The following cross-references are used in this glossary:

- *See* refers you from a nonpreferred term to the preferred term or from an abbreviation to the spelled-out form.
- *See also* refers you to a related or contrasting term.

A B C D E F G H I J K L M N O P Q R S T U V W X Z

A

activation key
: See license key.

advisory lock
: A type of lock that a process holds on a region of a file that signals any other process to not use or lock the region or an overlapping region. Other processes are not forced to comply.

AFM-S3 fileset
: A logical grouping of files in an Active File Management (AFM) enabled IBM Storage Scale system that connects to an external S3 bucket for data ingestion.

allocatable extent limit
: A maximum total capacity for the system. The allocatable extent limit is calculated from pool extent sizes.

allowlist
: A list of items, such as usernames, email addresses, or IP addresses, that are granted access to a certain system or function. When an allowlist is used for access control, all entities are denied access, except for those that are included in the allowlist.

application key
: See license key.

array
: An ordered collection, or group, of physical devices (disk drive modules) that are used to define logical volumes or devices. An array is a group of drives designated to be managed with a Redundant Array of Independent Disks (RAID).

asynchronous replication

A type of replication in which control is given back to the application as soon as the write operation is made to the source volume. Some time later, the write operation is made to the target volume. See also synchronous replication.

audit log
: An unalterable record of all commands or user interactions that are issued to the system.

authenticated user
: A user who has logged in to the system with a valid account (user ID and password).

authentication
: The mechanism by which a system determines what permissions a particular authenticated user has to access specific resources or actions. See also authorization.

authorization
: The mechanism by which a system determines what permissions a particular authenticated user has to access specific resources or actions. See also authentication.

authorization code
: An alphanumeric code generated for administrative functions, such as password resets or two-factor authentication bypass.

available capacity
: The amount of usable capacity that is not yet used in a system, pool, array, or MDisk.

B

bearer token
: A security token used in the HTTP Authorization header to authenticate API requests. In Content-Aware Storage, it can be obtained from OpenShift or an external identity provider (IDP).

block storage
: A unit of data storage on a device.

C

cache
: Storage or memory that is used to improve access times to instructions, data, or both. For example, data that resides in cache memory is normally a copy of data that resides elsewhere in slower, less expensive storage, such as on a disk or on another network node.

cache eviction
: A process by which data associated with a file is removed from the cache system. The data is removed either by using a Least Recently Used (LRU) algorithm when configured General Parallel File System (GPFS) hard or soft quota limits are exceeded or by issuing a command. When referenced again in the cache system, the data that is associated with the file is retrieved from the home system.

caching file system
: A storage mechanism that temporarily stores frequently accessed data to improve performance and reduce latency in data retrieval.

caching I/O group
: The I/O group in the system that performs the cache function for a volume.

call home
: A communication link established between a product and a service provider. The product can use this link to place a call to a service provider when it requires service. With access to the machine, service personnel can perform service tasks, such as viewing error and problem logs or initiating trace and dump retrievals.

capacity
: The amount of data that can be contained on a storage medium.

capacity recycling
: The amount of provisioned capacity that can be recovered without causing stress or performance degradation. This capacity identifies the amount of resources that can be reclaimed and provisioned to other objects in an environment.

capacity threshold
: The percent of total usable physical capacity that used capacity must exceed before a notification is sent. See also total usable physical capacity.

CasInstall custom resource
: A Kubernetes custom resource used to configure the installation parameters of the Content-Aware Storage (CAS) service, including enabling optional features like NVIDIA reranking.

CAS namespace
: A dedicated Kubernetes namespace where CAS components (such as pods, services, and configurations) are deployed. Deleting this namespace removes all CAS-related resources.

CAS operator
: A Kubernetes operator that automates the deployment, configuration, and lifecycle management of the CAS service within the OpenShift® environment.

CAS query user
: A user that is authorized to perform search queries against CAS domains. This includes users from Red Hat OpenShift or external identity providers, depending on the access control configuration.

certificate
: A digital document that binds a public key to the identity of the certificate owner, thereby enabling the certificate owner to be authenticated. A certificate is issued by a certificate authority and is digitally signed by that authority.

change volume
: A volume that is used in Global Mirror that holds earlier consistent revisions of data when changes are made.

child pool
: A user-defined capacity that is formed from capacity that is defined either in another pool or a system. See also parent pool.

CIFS
: See Common Internet File System.

CIM
: See Common Information Model.

CIM object manager (CIMOM)
: The common conceptual framework for data management that receives, validates, and authenticates the CIM requests from the client application. It then directs the requests to the appropriate component or service provider.

CIMOM
: See CIM object manager.

CKD
: See count key data.

CLI
: See command-line interface.

client
: A software program or computer that requests services from a server. See also host, server.

cloud account
: An agreement with a cloud service provider to use storage or other services at that service provider. Access to the cloud account is granted by presenting valid credentials.

cluster
: A group of computers and other resources that operate together as a single system.

command-line interface (CLI)
: A computer interface in which the input and output are text based.

Common Information Model (CIM)
: An implementation-neutral, object-oriented schema for describing network management or systems management information. The Distributed Management Task Force (DMTF) develops and maintains CIM specifications.

Common Internet File System (CIFS)
: A protocol that manages shared, remote file access for applications to files, printers, serial ports, and so on over a TCP/IP network.

compression
: A function that removes repetitive characters, spaces, strings of characters, or binary data from the data being processed and replaces characters with control characters. Compression reduces the amount of storage space that is required for data.

compute node
: An independent machine that contains one or more microprocessors, memory, storage, and network controllers and runs its own operating system and applications.

concurrent copy
: A function of the DFSMSdss component that is used to back up any collection of data at a point in time with minimum down time for the database or application that uses the collection of data.

ConfigMap
: An API object that holds key-value pairs of configuration data that can be consumed in pods or used to store configuration data. Pods can consume ConfigMaps as environment variables, command-line arguments, or as configuration files in a volume.

copyback
: A process that moves data back to its expected or preferred location to maintain an array in a more efficient configuration after a failed drive is replaced.

count-key-data record
: See data record.

count key data (CKD)
: 1. An architecture for a direct access storage device (DASD) device or logical device that specifies the access mechanisms for the logical data units on the device through a specific set of supported channel commands. Extensions to the CKD command set form the basis of Extended CKD.
  2. A data recording format that uses self-defining record formats in which each record on a volume is represented by up to three fields: a count field identifying the record and specifying its format, an optional key field that can be used to identify the data area contents, and an optional data field that typically contains the user data. See also data record, storage architecture type.

CRU
: See customer-replaceable unit.

customer-replaceable unit (CRU)
: An assembly or part that can be replaced in its entirety by a user when any one of its components fails.

custom resource
: A Kubernetes extension that is used to define and manage custom configurations for applications and services.

cylinder
: A unit of storage on a count-key-data (CKD) device with a fixed number of tracks.

D

data consistency
: A characteristic of the data at the target site where dependent write order is maintained to guarantee the recoverability of applications.

data lakehouse
: A unified data storage and processing architecture that combines the flexibility of a data lake with the structured querying and performance optimizations of a data warehouse, enabling scalable and efficient data analysis for AI and analytics applications.

data record
: A basic unit of data recording format. See also count key data, fixed-block architecture.

data reduction
: A set of techniques that can be used to reduce the amount of usable capacity that is required to store data. Examples of data reduction include data deduplication and compression. See also data reduction savings, stored capacity.

data reduction savings
: The total amount of usable capacity that is saved in a system, pool, or volume through the application of an algorithm such as compression or deduplication on the written data. This saved capacity is the difference between the written capacity and the used capacity. See also data reduction.

data source
: A storage location that hosts documents for use in AI workflows. Also see Creating a data source.

destage
: To move data from cache to a nonvolatile storage medium.

directory server
: A server that can add, delete, change, or search directory information on behalf of a client.

distributed RAID
: An alternative RAID scheme where the number of drives that are used to store the array can be greater than the equivalent, typical RAID scheme. The same data stripes are distributed across a greater number of drives, which increases the opportunity for parallel I/O and hence improves overall array performance. See also rebuild area.

DNS
: See Domain Name System.

domain
: A configuration that collects, transforms, and transfers data from multiple sources to a destination, enabling efficient data flow for analysis and AI models. Also see Creating a domain and connecting with data source.

Domain Name System (DNS)
: The distributed database system that maps domain names to IP addresses.

drive
: A data storage device. A drive can be either a magnetic disk drive or a solid-state drive (SSD).

drive class
: A combination of drive technology and speed, which uniquely defines a class of drives that have approximately the same performance characteristics.

drive technology
: A category of a drive that pertains to the method and reliability of the data storage techniques being used on the drive. Possible values include enterprise (ENT) drive, nearline (NL) drive, or solid-state drive (SSD).

E

ECKD
: See extended count key data.

effective capacity
: The amount of provisioned capacity that can be created in a system or pool without running out of usable capacity given the current data reduction savings being achieved. This capacity equals the usable capacity divided by the data reduction savings percentage.

enclosure
: The metal structure in which various electronic components are mounted.

encryption deadlock
: The inability to access encryption keys to decrypt data. See also encryption recovery key.

encryption key label
: The list of encryption key labels used by the storage system to identify keys that will be used on the key server.

encryption key manager
: See encryption key server.

encryption key server
: An internal or external system that runs a key manager that receives and then serves existing encryption keys or certificates to a storage system.

encryption recovery key
: An encryption key that allows a method to recover from an encryption deadlock situation where the normal encryption key servers are not available. See also encryption deadlock.

enterprise
: Pertaining to a type of data storage device that has higher error recovery limits, vibration tolerance, and end-to-end error detection than standard desktop hard drives.

extended count key data (ECKD)
: An extension of the count-key-data (CKD) architecture. It includes additional commands that can be used to improve performance.

extent type
: See storage architecture type.

F

failback
: The restoration of an appliance to its initial configuration after detection and repair of a failed network or component.

failover
: An automatic operation that switches to a redundant or standby system or node in the event of a software, hardware, or network interruption.

FAST API
: A high-performance Python web framework used to build RESTful APIs. In Content-Aware Storage, it powers the query service pod that handles search requests from client applications.

FBA
: See fixed-block architecture.

FC
: See Fibre Channel.

FC-AL
: See Fibre Channel Arbitrated Loop.

FCIP
: See Fibre Channel over IP.

FCP
: See Fibre Channel Protocol.

feature activation code
: See license key.

Fibre Channel (FC)
: A technology for transmitting data between computer devices. It is especially suited for attaching computer servers to shared storage devices and for interconnecting storage controllers and drives. See also zoning.

Fibre Channel Arbitrated Loop (FC-AL)
: An implementation of the Fibre Channel standards that uses a ring topology for the communication fabric; refer to American National Standards Institute (ANSI) INCITS 272-1996, (R2001). In this topology, two or more Fibre Channel end points are interconnected through a looped interface.

Fibre Channel connection (FICON)
: A Fibre Channel communication protocol designed for IBM mainframe computers and peripherals.

Fibre Channel extender
: A device used to extend a Fibre Channel link over a greater distance than is supported by the standard, usually a number of miles or kilometers. Devices must be deployed in pairs at each end of a link.

Fibre Channel over IP (FCIP)
: A network storage technology that combines the features of the Fibre Channel Protocol and the Internet Protocol (IP) to connect distributed SANs over large distances.

Fibre Channel Protocol (FCP)
: The serial SCSI command protocol used on Fibre Channel networks. See also open system.

FICON
: See Fibre Channel connection.

field-replaceable unit (FRU)
: An assembly that is replaced in its entirety when any one of its components fails.

file module

A component that provides file systems to network users. A file module must be provided with storage for the file systems.

fileset
: A logical subdivision of a file system used to organize and manage data independently within a larger storage environment.

file system (FS)
: A collection of files and certain attributes associated with those files.

file system storage
: Data storage that is organized into files and directories.

fixed-block architecture (FBA)
: An architecture for a virtual device that specifies the format of and access mechanisms for the virtual data units on the device. The virtual data unit is a block. All blocks on the device are the same size (fixed size). The system can access them independently. See also data record, storage architecture type.

fixed block
: See fixed-block architecture.

FlashCopy
: Pertaining to a point-in-time copy where a virtual copy of a volume is created. The target volume maintains the contents of the volume at the point in time when the copy was established. Any subsequent write operations to the source volume are not reflected on the target volume.

flash drive
: A data storage device, which is typically removable and rewritable, that uses solid-state memory to store persistent data. See also flash module.

flash module
: A modular hardware unit containing flash memory, one or more flash controllers, and associated electronics. See also flash drive.

flush-through mode
: See write-through mode.

form factor
: The industry-standard physical dimensions of a storage system drive enclosure. Possible values include “3.5 inch”, “2.5 inch”, and “1.8 inch.”

frame
: The hardware support structure, covers, and all electrical parts mounted therein that are packaged as one entity for shipping.

FRU
: See field-replaceable unit.

FS
: See file system.

full restore operation
: A copy operation where a local volume is created by reading an entire a volume snapshot from cloud storage.

full snapshot
: A type of volume snapshot that contains all the volume data. When a full snapshot is created, an entire copy of the volume data is transmitted to the cloud.

G

General Parallel File System (GPFS)
: A high-performance shared-disk file system that can provide data access from nodes in a clustered system environment.

Global Mirror
: A method of an asynchronous replication that maintains data consistency across multiple volumes within or across multiple systems. Global Mirror is generally used where distances between the source site and target site cause increased latency beyond what the application can accept.

GPFS
: See General Parallel File System.

GPU resource indicator
: A Kubernetes resource identifier (for example, nvidia.com/gpu, nvidia.com/mig-7g.80gb) used to specify the type and configuration of GPU resources allocated to a container or service.

Granite Embedding model
: One of a collection of embedding models that generate high-quality text embeddings and a reranker model to improve the relevance and quality of search results or recommendations.

group id (GID)
: In the UNIX or Linux operating system, an integer that uniquely identifies each group of users to the operating system.

H

Hardware Management Console (HMC)
: A system that controls managed systems, including the management of logical partitions and use of Capacity Upgrade on Demand. Using service applications, the HMC communicates with managed systems to detect and consolidate information, which can then be sent for analysis.

HMC
: See Hardware Management Console.

host
: A physical or virtual computer system that hosts computer applications, with the host and the applications using storage. See also client, host, server.

host cluster
: A configured set of physical or virtual hosts that share one or more storage volumes to increase scalability or availability of computer applications.

host interface card
: See interface card.

host object
: A logical representation of a host within a storage system that is used to represent the host for configuration tasks.

hot-spare
: Pertaining to redundant hardware (such as an adapter, a disk, a drive, or a server) that is installed and available in the event of a hardware failure.

hybrid search
: A search approach that combines semantic and keyword search methods to improve accuracy and relevance.

HyperSwap
: Pertaining to a function that provides continuous, transparent availability against storage errors and site failures, and is based on synchronous replication.

I

identity provider (IDP)

A service that creates, maintains, and manages identity information and is the source of identities that are communicated to service providers in federated relationships.

IDP token
: A secure token issued by an external identity provider (IDP) used to authenticate users when accessing CAS services, including semantic search APIs.

I/O
: See input/output.

I/O enclosure
: A hardware unit in a storage system where data is transferred into and out of the system.

incremental restore operation
: A copy operation where a local volume is modified to match a volume snapshot by reading from cloud storage only the parts of the volume snapshot that differ from the local volume.

incremental snapshot
: A type of volume snapshot where the changes to a local volume relative to the volume's previous snapshot are stored on cloud storage.

input/output (I/O)
: Pertaining to a device, process, channel, or communication path involved in data input, data output, or both.

interface card
: An optional part of a node canister that provides the system with additional host and storage connectivity options.

interface node
: A node that connects a system to an Internet Protocol (IP) network for file-serving capabilities by using service protocols.

Internet Small Computer System Interface (iSCSI)
: An IP-based standard for linking data storage devices over a network and transferring data by carrying SCSI commands over IP networks. See also Small Computer System Interface.

iSCSI
: See Internet Small Computer System Interface.

J

junction path
: A mount point in IBM Storage Scale that connects a fileset to the file system hierarchy, allowing access to stored data.

K

Kafka
: An open-source distributed event streaming platform used for building real-time data pipelines and streaming applications. In Content-Aware Storage, Kafka is used to manage messaging and data flow between components.

Kafka broker
: A server within a Kafka cluster that stores and distributes messages to consumers. It handles incoming data streams and ensures reliable delivery.

Kafka topic
: A category or feed name used in Apache Kafka to which records are sent and from which consumers read.

key server
: A server that negotiates the values that determine the characteristics of a dynamic virtual private network (VPN) connection that is established between two endpoints.

keyword search
: A traditional search method that retrieves results based on exact word or phrase matches.

L

large language model (LLM)
: A language model with many parameters, trained on a large quantity of text.

licensed capacity
: The amount of capacity on a storage system that a user is entitled to configure.

license key
: An alphanumeric code that activates a licensed function on a product.

license key file
: A file that contains one or more licensed keys.

M

machine signature
: A string of characters that identifies a system. A machine signature might be required to obtain a license key.

management node
: A node that is used for configuring, administering, and monitoring a system.

maximum replication delay
: The number of seconds that Metro Mirror or Global Mirror replication can delay a write operation to a volume.

Metro Global Mirror
: A cascaded solution where Metro Mirror synchronously copies data to the target site. This Metro Mirror target is the source volume for Global Mirror that asynchronously copies data to a third site. This solution has the potential to provide a disaster recovery with no data loss at Global Mirror distances when the intermediate site does not participate in the disaster that occurs at the production site.

Metro Mirror
: A method of synchronous replication that maintains data consistency across multiple volumes within the system. Metro Mirror is generally used when the write latency caused by the distance between the source site and target site is acceptable to application performance.

N

natural language processing (NLP)

A field of artificial intelligence and linguistics that studies the problems inherent in the processing and manipulation of natural language, with an aim to increase the ability of computers to understand human languages.

nearline
: Pertaining to a type of storage in which data is available in a short amount of time, but not instantly.

nearline SAS drive
: A drive that combines the high capacity data storage technology of a Serial Advanced Technology Attachment (SATA) drive with the benefits of a serial-attached SCSI (SAS) interface for improved connectivity.

node
: A single processing unit within a system. For redundancy, multiple nodes are typically deployed to make up a system.

NVIDIA NIM pipeline
: A pipeline configuration that uses NVIDIA's NeMo Inference Microservices (NIM) to process and extract insights from data, typically integrated with IBM Storage Scale for AI workloads.

O

open system
: A system that complies with industry-defined interoperability standards. An open system can be connected to other systems complying with the same standards. See also Small Computer System Interface, Fibre Channel Protocol.

order confirmation code
: See authorization code.

overhead capacity
: An amount of usable capacity that is occupied by metadata in a system or pool and other data that is used for system operations.

overprovisioned ratio
: The ratio of provisioned capacity to usable capacity in a system or pool.

overprovisioning
: The result of creating more provisioned capacity in a storage system or pool than there is usable capacity. Overprovisioning occurs when thin provisioning or data reduction techniques ensure that the used capacity of the provisioned volumes is less than their provisioned capacity.

P

parent pool
: A storage pool that receives its capacity from MDisks and has, or will have, some of its capacity allocated to child pools. See also child pool.

performance group
: A collection of volumes that is assigned the same performance characteristics. See also performance policy.

performance policy
: A policy that specifies performance characteristics, for example quality of service (QoS). See also performance group.

PFC
: See priority flow control.

pod
: A group of containers that are running on a Kubernetes cluster. A pod is a runnable unit of work, which can be a either a stand-alone application or a microservice.

pool
: See storage pool.

pool pair
: Two storage pools that are required to balance workload. Each storage pool is controlled by a separate node.

port
: The physical entity within a host, system, or storage system that performs the data communication (transmitting and receiving) over the Fibre Channel.

priority flow control (PFC)
: A link-level flow control mechanism, IEEE standard 802.1Qbb. PFC operates on individual priorities. Instead of pausing all traffic on a link, PFC is used to selectively pause traffic according to its class.

projected capacity
: The estimated volume capacity that is available for volume creation, given the current average performance of any data compression, excluding thin-provisioning savings. See also thin-provisioning savings.

protocol
: A set of rules controlling the communication and transfer of data between two or more devices or systems in a communication network.

provisioned capacity
: The total capacity of all volumes and volume copies in a system or pool.

persistent volume (PV)
: Networked storage in a cluster that is provisioned by an administrator.

persistent volume claim (PVC)
: A request for cluster storage. It allows a pod to use a persistent volume (PV) by claiming a portion of it.

Q

query service pod
: A Kubernetes pod within the CAS namespace that exposes the search API through FAST API, enabling client applications to query the vector database.

R

rack
: A free-standing structure that can hold multiple servers, storage systems, chassis, switches, and other devices.

RAID
: See Redundant Array of Independent Disks.

RAID 0
: A data striping technique, which is commonly called RAID Level 0 or RAID 0 because of its similarity to common, RAID, data-mapping techniques. It includes no data protection, however, so, strictly speaking, the appellation RAID is a misnomer. RAID 0 is also known as data striping.

RAID 1
: A form of storage array in which two or more identical copies of data are maintained on separate media.

RAID 10
: A collection of two or more physical drives that present to the host an image of one or more drives. In the event of a physical device failure, the data can be read or regenerated from the other drives in the RAID due to data redundancy.

RAID 5
: A form of parity RAID in which the disks operate independently, the data stripe size is no smaller than the exported block size, and parity check data is distributed across the array's disks.

RAID 6
: A form of RAID that can continue to process read and write requests to all of an array's virtual disks in the presence of two concurrent disk failures.

RAID level
: The level of protection provided by the specific techniques of striping, mirroring, or parity used by a Redundant Array of Independent Disks (RAID).

RAID type
: See RAID level.

raw capacity
: The reported capacity of the drives in the system before formatting or RAID (Redundant Array of Independent Disks) is applied.

rebuild area
: Reserved capacity that is distributed across all drives in a redundant array of drives. If a drive in the array fails, the lost array data is systematically restored into the reserved capacity, returning redundancy to the array. The duration of the restoration process is minimized because all drive members simultaneously participate in restoring the data. See also distributed RAID.

reclaimable capacity
: The amount of provisioned capacity that can be recovered without causing stress or performance degradation. This capacity identifies the amount of resources that can be reclaimed and provisioned to other objects in an environment.

reclaimed capacity
: See reclaimable capacity.

recovery key
: See encryption recovery key.

Redundant Array of Independent Disks (RAID)
: A collection of two or more physical disk drives that present to the host an image of one or more logical disk drives. In the event of a physical device failure, the data can be read or regenerated from the other disk drives in the array due to data redundancy.

repo
: See repository.

repository (repo)
: A persistent storage area for data and other application resources.

reranker
: A specialized AI model that is designed to improve the relevance and accuracy of search and retrieval results, especially in retrieval-augmented generation (RAG) pipelines.

reranking
: A post-processing step in search workflows that reorders retrieved results based on additional criteria or AI models to improve relevance.

reserved capacity
: The amount of used capacity that is made up of capacity reserved for system use. See also total usable physical capacity.

retrieval augmented generation (RAG)
: A technique in which a large language model is augmented with knowledge from external sources to generate text. In the retrieval step, relevant documents from an external source are identified from the user's query. In the generation step, portions of those documents are included in the LLM prompt to generate a response grounded in the retrieved documents.

S

SCIM client
: A configuration block in the `IdentityProvider` custom resource that enables CAS to retrieve user and group information from the IDP using the SCIM (System for Cross-domain Identity Management) protocol.

SCSI
: See Small Computer System Interface.

SCSI-FCP
: See SCSI Fibre Channel Protocol.

SCSI device
: A product, such as a drive or adapter, connected to a host through an I/O interface using the Small Computer System Interface (SCSI) protocol. A SCSI device is either an initiator,target, or both. See also Small Computer System Interface.

SCSI Fibre Channel Protocol (SCSI-FCP)
: A standard that defines the protocol used to transfer Small Computer System Interface (SCSI) commands over the transport physical layer of the Fibre-Channel interface. This standard is published by ANSI as X3.269-1996.

SCSI initiator
: The system component that initiates communications with attached targets.

SCSI target
: A device that acts as a subordinate to a SCSI initiator and consists of a set of one or more logical units (LUs), each with an assigned logical unit number (LUN). The LUs on the SCSI target are typically I/O devices.

search API
: An application programming interface that enables external applications to perform semantic, keyword, or hybrid searches on ingested content.

security context constraint (SCC)
: A Kubernetes policy object used to define permissions and capabilities for pods, such as allowing containers to run with elevated privileges like SETUID and SETGID.

semantic search
: A search technique that uses the meaning of words and phrases rather than exact keyword matches to retrieve relevant results.

server
: A computer program or a device that provides functions for other programs or devices, called clients. See also client, host.

Server Message Block (SMB)
: A protocol that manages requests and responses in a client/server environment so that clients on a network can share files, directories, and devices.

Small Computer System Interface (SCSI)
: An ANSI-standard electronic interface that allows personal computers to communicate with peripheral hardware, such as disk drives, tape drives, CD-ROM drives, printers, and scanners faster and more flexibly than previous interfaces. See also open system, Internet Small Computer System Interface, SCSI device.

SMB

See Server Message Block.

solid-state drive (SSD)

1. A storage device that contains nonvolatile flash memory. A solid-state drive (SSD) has no moving mechanical components.

space
: See capacity.

space efficient
: See thin provisioning.

spare drive
: A drive reserved in an array for rebuilding a failed drive in a RAID. Should a drive fail in a RAID, a spare drive from within that device adapter (DA) pair will be selected to rebuild it.

SSD
: See solid-state drive.

standard-provisioned volume
: A volume that completely uses storage at creation.

standard provisioning
: The ability to completely use a volume's capacity for that specific volume.

storage architecture type (storage type)
: The type of storage architecture, either count key data (CKD) or fixed block (FB), for which an array, pool, or volume is provisioned. See also count key data, fixed-block architecture.

storage class
: A storage class is a Kubernetes resource that defines different types of storage (for example, performance and replication) available in a cluster.

storage enclosure
: A specialized chassis that is designed to hold and power drives while providing a mechanism to allow them to communicate to one or more separate computers.

storage node
: A component of a storage system that provides internal storage or a connection to one or more external storage systems.

storage pod
: A subcomponent of a network-attached storage (NAS) system that consists of two or more storage nodes and one or more supported storage systems.

storage pool (pool)
: A collection of storage that identifies an underlying set of resources. These resources provide the capacity and management requirements for a volume or set of volumes.

storage system
: A system that provides persistent storage within a network. A storage system can include facilities for host attachment, user role authentication, a command-line interface (CLI), a graphical user interface (GUI), and storage devices that most often include Redundant Array of Independent Disks (RAID) controllers. It might also include agents for enabling third-party management software to monitor or manage the storage devices.

storage type
: See storage architecture type.

stored capacity
: The amount of capacity that is used to store data that is written by a host after data reduction See also data reduction, total usable physical capacity.

support assistance
: A function that is used to provide support personnel access to the system to complete troubleshooting and maintenance tasks.

synchronous replication
: A type of replication in which the application write operation is made to both the source volume and target volume before control is given back to the application. See also asynchronous replication.

syslog
: A standard for transmitting and storing log messages from many sources to a centralized location to enhance system management.

T

thin-provisioned volume
: A volume that allocates storage when data is written to it.

thin-provisioning savings
: The total amount of usable capacity that is saved in a system, pool, or volume by consuming usable capacity only when needed as a result of write operations. The capacity that is saved is the difference between the provisioned capacity minus the written capacity. See also volume capacity, projected capacity, written capacity.

thin provisioning
: The ability to defer capacity allocation on a storage resource until data is actually written to it.

time slicing
: A GPU resource management technique that allows multiple workloads to share a GPU by allocating time slots for execution, improving utilization.

TLS secret
: A Kubernetes secret that stores TLS certificates and keys used to secure communication with external services, such as S3 buckets.

tokenizer
: A text segmentation program that scans text and determines if and when a series of characters can be recognized as a token.

total capacity savings
: The total amount of usable capacity that is saved in a system, pool, or volume through thin provisioning and data reduction techniques. This saved capacity is the difference between the used usable capacity and the provisioned capacity.

total usable physical capacity
: The amount of physical configured storage space that is available for stored capacity or reserved capacity. This capacity can consist of both internal storage through arrays and external storage through MDisks. See also reserved capacity, capacity threshold, stored capacity.

transparent cloud tiering
: The functions that use cloud storage as an extension of on-premises storage.

trial license
: A temporary entitlement to use a licensed function.

TSE for FlashCopy
: A thin-provisioning method in which storage space is allocated from a TSE repository on an as needed basis. See also TSE repository.

TSE repository
: The amount of capacity in a storage pool reserved for volumes that use a thin-provisioning method of TSE for FlashCopy. See also TSE for FlashCopy.

U

unmapped volume capacity
: The amount of volume capacity that is not mapped to a host. See also volume capacity.

update
: 1. Software maintenance such as a manufacturing refresh, refresh pack, or fix pack that changes the modification level of a product.
  2. To modify a file or data set with current information.
  3. To apply fixes to a system.

upgrade
: 1. Any hardware or software change to a later release, or any hardware addition or software addition.
  2. To install a new version or release of a product to replace an earlier version or release of the same product.

usable capacity
: The amount of capacity that is provided for storing data on a system, pool, array, or MDisk after formatting and RAID techniques are applied.

used capacity
: The amount of usable capacity that is taken up by data or overhead capacity in a system, pool, array, or MDisk after data reduction techniques have been applied.

user role
: An identifier that is assigned to a user that defines the set of permissions that are granted to that user.

V

vector database
: A specialized database within the CAS namespace that stores vector embeddings generated from ingested documents, enabling semantic search and retrieval.

vector embedding
: A numerical representation of text data that is used to capture semantic meaning, enabling similarity-based search and retrieval in vector databases.

virtual capacity
: See provisioned capacity.

virtualized capacity
: The amount of capacity that is contributed to a storage pool by a given provisioning group.

virtual machine (VM)
: An emulation of a particular computer system. Virtual machines operate based on the computer architecture and functions of a real or hypothetical computer. Their implementations might involve specialized hardware, software, or a combination of both.

VM
: See virtual machine.

volume
: A fixed amount of physical or virtual storage on a data storage medium.

volume access set
: The set of I/O groups that allows host access to a volume. This set can optionally include the caching I/O group.

volume capacity
: The total capacity for all volumes in a system or storage pool. Volume capacity is defined by the client when a volume is created and surfaced to the host. See also thin-provisioning savings, unmapped volume capacity.

volume snapshot
: A collection of objects on a cloud storage account that represents the data of a volume at a particular time.

W

watch folder
: A designated directory in a file system that is monitored for changes, triggering automated processing workflows when new or modified files are detected.

worldwide ID (WWID)
: A name identifier that is unique worldwide and that is represented by a 64-bit value that includes the IEEE-assigned organizationally unique identifier (OUI).

worldwide name (WWN)
: A 64-bit, unsigned name identifier that is unique.

worldwide node name (WWNN)
: A unique 64-bit identifier for a host containing a Fibre Channel port. See also worldwide port name.

worldwide port name (WWPN)
: A unique 64-bit identifier associated with a Fibre Channel adapter port. The WWPN is assigned in an implementation-independent and protocol-independent manner. See also worldwide node name.

write-through mode
: A process in which data is written to a storage device at the same time as the data is cached.

written capacity
: The amount of usable capacity that would have been used to store written data in a system or pool if data reduction was not applied. See also thin-provisioning savings.

written capacity limit
: The largest amount of capacity that can be written to a drive, array, or MDisk. The limit can be reached even when usable capacity is still available.

WWID
: See worldwide ID.

WWN
: See worldwide name.

WWNN
: See worldwide node name.

WWPN
: See worldwide port name.

X

X.509 certificate
: A certificate that contains information that is defined by the X.509 standard.

Z

z Global Mirror
: A method of an asynchronous replication function that maintains data consistency across multiple volumes that are attached to a z/OS system. Time-based data consistency is maintained through the Data Facility Storage Management Subsystem (DFSMS) system data mover (SDM) component.

zoning
: The grouping of multiple ports to form a virtual, private, storage network. Ports that are members of a zone can communicate with each other, but are isolated from ports in other zones. See also Fibre Channel.

Notices

Legal notices, copyright information, and terms governing the use of IBM Fusion product documentation.

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
(your company name) (year)
Portions of this code are derived from IBM Corp.
Sample Programs. © Copyright IBM Corp. _enter the year or years_.

- **EU Data Act (Regulation 2023/2854)**
  EU Data Act (Regulation 2023/2854) compliance information describes the types of product data generated by storage products, including vital product data, configuration data, and logging data, along with their retention, access, and control characteristics.

Programming interface information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of IBM Fusion. This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM Fusion. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking: Programming Interface information.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name and position for purposes of session management, authentication, enhanced user usability, and single sign-on configuration. These cookies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users through cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Safety and environmental notices

For more information, see Safety and environmental notices.

EU Data Act (Regulation 2023/2854)

EU Data Act (Regulation 2023/2854) compliance information describes the types of product data generated by storage products, including vital product data, configuration data, and logging data, along with their retention, access, and control characteristics.

Overview

This Data Information Page provides information regarding data generated by this product. To the extent that access to and use of this data is governed by Regulation (EU) 2023/2854 ("Data Act") the EU Data Act Terms available at this link apply.

Vital Product Data (VPD) / Machine Readable Product Data (MRPD)

Description

Human readable information on product data including Serial Number, Machine Type and Model, Date of manufacture, and similar identifying information.

Data characteristics

- Format: Text files, CSV files, HTTP/HTTPS, JSON or XML files, and other human readable file types
- Volume: Varies by product, but VPD/MRPD size does not typically change; as of 2025, this data could be up to 4 GB
- Generation: Real-time when the product is powered on; most VPD/MRPD data remains static

Storage and retention

- Storage products can send product data to IBM, sometimes through a service or management console
- Products and consoles can store product data sent to IBM temporarily, deleting data based on space management requirements and/or time
- IBM servers can store product data for at least 180 days

User access and control

- Product data can be obtained by customers through GUIs, CLIs, SNMP, REST interfaces, host commands, and other mechanisms
- Users cannot erase VPD/MRPD data
- Access available through:
  - Service console GUI, CLI and/or REST interface
  - Product-specific interfaces (GUIs, CLIs, REST interface, SNMP, host commands)

Configuration Data

Description

Human readable information describing the configuration of the product, identifying parts such as FRUs (Field Replaceable Units) and Feature Codes.

Data characteristics

- Format: Text files, CSV files, HTTP/HTTPS, JSON or XML files, and other human readable file types
- Volume: Varies by product and is expected to change with future code product updates and as new features are added; as of 2025, this data could be up to 4 GB
- Generation: Real-time when the product is powered on; most configuration/inventory data is stable until the customer upgrades the product

Storage and retention

- Storage products can send product data to IBM, sometimes through a service or management console
- Products and consoles can store product data sent to IBM temporarily, deleting data based on space management requirements and/or time
- IBM servers can store product data for at least 180 days

User access and control

- Product data can be obtained by customers through GUIs, CLIs, SNMP, REST interfaces, host commands, and other mechanisms
- Access available through:
  - Service console GUI, CLI and/or REST interface
  - Product-specific interfaces (GUIs, CLIs, REST interface, SNMP, host commands)

Logging Data

Description

Files containing data generated by system use and available to customers, including:

- High level error logs tracking error information
- Command logs showing information such as read or write commands sent to the storage device
- Access logs tracking users logging into the system and running admin commands

Data characteristics

- Format: Text files, CSV files, HTTP/HTTPS, JSON or XML files, and other human readable file types
- Volume: Varies by product and is expected to change with future code product updates and as new features are added; as of 2025, this data could be up to 4GB
- Generation: Real-time when the product is powered on; logging runs continuously and is finite in length, with older data removed as newer data is generated

Storage and retention

- Storage products can send product data to IBM, sometimes through a service or management console
- Products and consoles can store product data sent to IBM temporarily, deleting data based on space management requirements and/or time
- IBM servers can store product data for at least 180 days

User access and control

- Product data can be obtained by customers through GUIs, CLIs, SNMP, REST interfaces, host commands, and other mechanisms
- Access available through:
  - Service console GUI, CLI and/or REST interface
  - Product-specific interfaces (GUIs, CLIs, REST interface, SNMP, host commands)