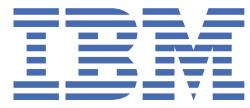


IBM Storage Fusion Software



Tables of Contents

IBM Storage Fusion documentation	1
What's new	1
Hotfixes	2
Product overview	2
Benefits of IBM Storage Fusion	3
IBM Storage Expert Care	3
IBM Storage Fusion trial version	3
Fusion Essentials	3
Security in IBM Storage Fusion	4
Verifying image signatures	4
Deploying IBM Storage Fusion	5
Prerequisites	5
Obtaining entitlement key	6
Creating image pull secret	6
Creating image pull secret for IBM Cloud based installation	7
System requirements	9
SecurityContextConstraints	11
Enterprise registry for IBM Storage Fusion installation	11
Mirroring your images to the enterprise registry	11
Mirroring Red Hat operator images to enterprise registry	13
Mirroring IBM Storage Fusion images	14
Mirroring IBM Storage Scale images	15
Mirroring Data Foundation images deployed on OpenShift Container Platform version 4.14 or 4.15	17
Migrating your Data Foundation ImageContentSourcePolicy to ImageDigestMirrorSet	19
Mirroring Backup & Restore images	20
Mirroring Data Cataloging images	22
Installing IBM Storage Fusion on On-premises VMware	24
Installing IBM Storage Fusion on On-premises Bare Metal	25
Installing IBM Storage Fusion on On-premises Linux on IBM zSystems and zCX	26
Installing IBM Storage Fusion on IBM Cloud	27
Installing IBM Storage Fusion on Amazon Web Services	28
Installing IBM Storage Fusion on Microsoft Azure	29
Installing IBM Storage Fusion on On-premises IBM Power Systems	30
Installing IBM Storage Fusion on Google Cloud	31
Installing IBM Storage Fusion from enterprise registry	31
Installing IBM Storage Fusion On-premises	32
Validating IBM Storage Fusion installation	33
Uninstalling IBM Storage Fusion	41
Deploying IBM Storage Fusion services	42
IBM Storage Fusion Services support matrix	43
Installing services	43
Deploying services from OpenShift Container Platform	44
Global Data Platform	45
Configuring Global Data Platform service on infrastructure nodes	46
Data Foundation	47
Data Foundation	47
Backup & Restore	48
Backup & Restore hub	49
Backup & Restore spoke	50
Establishing connection between hub and spoke	51
Disabling the connection	51
Data Cataloging	52
Uninstalling services	53
Disabling connections	53
Uninstalling Data Cataloging	53
Uninstalling Backup & Restore	54
Uninstalling Data Foundation	54
Uninstalling Global Data Platform	55
Upgrading IBM Storage Fusion	56
Prerequisites for enterprise registry upgrade	57
Upgrading IBM Storage Fusion services	58
Upgrading Red Hat OpenShift Data Foundation 4.12 to IBM Storage Fusion Data Foundation 4.12 or higher	60
Deploying your workloads	61
Deploying your workloads with Persistent volumes	61
Static provisioning of Persistent Volume in filesystem	62
Dynamic provisioning of Persistent Volume in filesystem	62
Storage provisioning using Container Storage Interface driver	64
Configuring Data Foundation storage	64
Configuring Data Foundation local storage	65

Adding disks to storage nodes	66
Adding nodes to your IBM Fusion Data Foundation storage	66
Configuring local stretch cluster DR with Fusion Data Foundation	67
Configuring Data Foundation dynamic storage	68
Adding capacity to storage nodes	69
Adding nodes to your IBM Fusion Data Foundation storage	69
Configuring Data Foundation in external mode	70
Setting up encryption for storage	71
Configuring encryption for Global Data Platform storage	71
Preparing to connect to an external KMS server in IBM Fusion Data Foundation	72
Enabling encryption with the token authentication using HashiCorp Vault(manual part)	73
Enabling encryption with the Kubernetes authentication using HashiCorp Vault (manual part)	73
Enabling encryption using Thales CipherTrust Manager (manual part)	75
Accessing remote IBM Storage Scale storage cluster in IBM Storage Fusion	75
Connecting to remote IBM Storage Scale file systems	76
Applications	78
Managing the application details	79
Data protection	80
Achieving data consistency	81
Backup and Restore	81
Hub and spoke model overview	82
FIPS-140-2	83
Prerequisites	83
Backup & restore configuration parameters	83
Backup & Restore Block PVC Technical preview	85
Reconnecting OpenShift Container Platform cluster	85
Overview	86
Connecting and managing cluster connections	87
Protecting your applications	88
Backing up applications	88
Backup storage locations	89
Creating a secret	89
Adding a backup storage location	90
Using MCG or Noobaa as a backup storage location	91
Using IBM Storage Protect as a backup storage location	91
Backup policies	92
Creating backup policy	93
Setting up backup policies from Applications list page	94
Selecting applications for protection	94
Backup now	95
Managing backup policy	95
Jobs	95
Jobs details	96
Managing applications backups	97
Restoring an application	98
Service protection	99
Configuring service backups	99
Recovering service backups	100
Backup and restore commands	101
Monitoring and managing Backup & restore service from OpenShift Container Platform	104
Orchestrate a backup or restore	107
Custom backup and restore workflows	108
Creating a Recipe	108
Assigning a Recipe	111
Backup and restore of IBM Cloud Pak for Data to the same or different cluster	112
IBM Storage Fusion repository for recipes	113
Data Cataloging	113
Data Cataloging architecture	114
Role-based access control	116
Data source connections	117
Cataloging metadata	117
Enriching metadata	117
Policy engine	118
Applications	118
Graphical user interface	119
Reports for Data Cataloging	120
Planning	120
Data migration path from 2.0.4 OVA environment	120
Backing up data	121
Restoring data	121
Configuring Data Cataloging	122
Setting up Data Cataloging custom user and password	122
Configure data source connections	122

IBM Storage Scale data source connections	123
IBM Storage Scale scanning considerations	123
Security considerations	123
Performance considerations	124
Prerequisites for automated scanning	124
Identify the IBM Storage Scale cluster and node list	124
Creating or identifying a user ID and password for scanning	124
Validate scan user permissions and configuration	125
Check the dependencies for optimized scanning	125
Creating an IBM Storage Scale data source connection	126
Automated scanning of an IBM Storage Scale data source	127
Automated scanning of an IBM Storage Scale filesset	127
Manual scanning of an IBM Storage Scale data source	128
Performing file system scan to collect metadata from IBM Storage Scale	129
Copying the output of the IBM Storage Scale file system scan to the IBM Spectrum Discover master node	130
Ingesting metadata from IBM Storage Scale file system scan in IBM Spectrum Discover	131
Ingesting quota information from the file system	131
IBM Storage Archive data source connections	131
IBM Cloud Object Storage data source connection	132
Prerequisites	132
Creating an IBM Cloud Object Storage data source connection	132
Scanning an IBM Cloud Object Storage data connection	134
Best practices for scanning IBM Cloud Object Storage systems	136
Enabling bucket notifications for Ceph Object Storage	136
Enabling bucket notifications for Ceph Object Storage on Red Hat OpenShift	138
Replaying IBM Cloud Object Storage notifications	139
Overview of architecture	139
Configuration file	140
Replay performance	144
Replay tasks and vault settings	145
Including and excluding vaults	146
Stats files	147
Replay	147
Initialization for Replay	147
Error conditions	147
Output	147
Renaming a vault for Replay	148
Starting the Replay	148
Debug mode for Replay	148
Notifier	149
Limitations	149
Starting the Notifier	149
Notifier operation	149
Stopping the Notifier	150
Restarting the Notifier	150
Progress report	150
Logging	152
IBM Cloud Object Storage Scanner output data	153
Appendix	155
Configure IBM Cloud Object Storage notifications for Data Cataloging	160
Enabling IBM Cloud Object Storage notification services	161
Authenticating, encrypting, and enabling	161
Authenticating, encrypting, and enabling services on Red Hat OpenShift	162
Testing the IBM Cloud Object Storage notification service	163
Monitoring the IBM Cloud Object Storage access logs	164
Monitoring the Data Cataloging producer IBM Cloud Object Storage logs	164
Monitoring the IBM Spectrum Discover dashboard for IBM Cloud Object Storage ingestion	164
Data Cataloging and S3 object storage data source connections	165
Creating an S3 object storage data connection	165
Scanning an S3 object storage data connection	167
Best practices for scanning an S3 object storage system	169
Scanning an Elastic Storage Server data source connection	169
Creating a Network File System data source connection	170
Creating an SMB data source connection	170
Creating an IBM Storage Protect data source connection	171
Configuring data source connections offline	172
Editing and using the TimeSinceAccess and Size Range buckets	172
Using custom TLS certificate	174
IBM Spectrum Storage software requirements	174
Data Cataloging installation with alternative VLAN	175
Managing user access	177
Initial login	178

Resetting the sdadmin password	182
Password policies	182
Changing password	183
Managing user accounts	183
Creating user accounts	184
Managing groups	185
Creating groups	185
Managing collections	186
Creating collections	187
Managing LDAP and IBM Cloud Object Storage System connections	187
Creating an LDAP connection	188
Configuring self-signed certificates to secure LDAPS connection	189
Creating an IBM Cloud Object Storage connection	189
Managing metadata policies	190
Adding policies	191
Adding auto-tagging policy parameters	192
Adding deep-inspection policy parameters	192
Adding content search policy parameters	192
Running policies	193
Viewing policies	193
Viewing policy history details and logs	194
Modifying policies	195
Deleting policies	195
Using content search policies	196
Identifying the required regex expressions	196
Creating a content search policy	197
Viewing content search application logs	197
Hints and tips for using content search	198
Tiering data by using ScaleILM application	198
Viewing ScaleILM application logs	200
Copying data by using ScaleAFM application	201
Viewing ScaleAFM application logs	202
Exporting metadata to IBM Watson Knowledge Catalog	202
Configuring Watson Knowledge Catalog connector	203
Configuring Watson Knowledge Catalog using the utility script	203
Configuring Watson Knowledge Catalog manually	203
Exporting metadata	204
Mapping similar source connections in Watson Knowledge Catalog	205
Troubleshooting export issues	206
Authentication failure with Watson Knowledge Catalog - both on-Premise and IBM Cloud	206
Incorrect WKC URL configuration	206
Invalid connection type in catalog	206
No linked connection type	207
WKC connector pod in CrashLoopBackoff state	207
S3 connection issues	208
Importing externally curated tags for COS/S3 using import tags application	208
Initiating Policy Using REST API	209
Registering import tags application	209
Defining the import tag policy type	210
Viewing the policy status	210
Viewing the import tags application log	211
Performing retention analytics on IBM Storage Protect archive data	211
Enabling the retention analytics feature	212
Adding IBM Storage Protect data source and initiating scan	213
Searching expiry time	214
Managing tags	215
Creating tags	215
Viewing and searching tags	216
Editing tags	216
Deleting tags	217
Discover data	217
Searching	217
Grouping data by file type	218
Searching system and custom metadata fields	218
System metadata fields to search on	219
Access control list metadata to search on	219
How to use ACL data in policies	221
Search on custom metadata fields	222
Examples of search filters	223
Search results table	224
Refine search results	224
Sort search results	224
Tag search results manually	224
Managing applications	224
Using the IBM Spectrum Discover application catalog	225
Creating your own applications to use in the Data Cataloging application catalog	227

Reports	227
High availability for a Db2 Warehouse MPP deployment	228
Reintegrating a failed data node into an IBM Db2 Warehouse MPP cluster	230
Monitoring data sources	230
Viewing data source status	231
Viewing data source connections	234
Recommended to move	234
Deleting or editing a connection	236
Monitoring the Data Cataloging service environment	238
Audit log	238
Configurable log trimmer handler	239
SD-Monitor	239
Updating the network configuration	239
Using a third-party data movement application to manage data	240
Data movement with IBM Spectrum Discover and Moonwalk	241
Preserving tags during data movement	242
Enabling feature for skipping the snapshot directories	242
Enabling skip snapshot directories feature on Red Hat OpenShift	243
Data protection	244
Backup and restore	244
Backup and restore scripts	244
Disaster recovery procedures	244
Preparations for disaster recovery	245
Running disaster recovery for Red Hat OpenShift	245
REST API for Data Cataloging	245
Data Cataloging REST APIs	245
API endpoints	246
Asynchronous endpoints	246
Status codes	247
Authentication process	247
Endpoints for working with a Db2 Warehouse	248
/db2whrest/v1/search: POST	248
/db2whrest/v1/report: POST	252
/db2whrest/v1/report: GET	258
/db2whrest/v1/report/<report_id>: GET	258
/db2whrest/v1/report/<report_id>/download: GET	259
/db2whrest/v1/bucket: GET	260
/db2whrest/v1/buckets/<bucket>: GET	261
/db2whrest/v1/report/<report_id>: PUT	261
/db2whrest/v1/buckets/<bucket>: PUT	262
/db2whrest/v1/report/<report_id>: DELETE	263
/db2whrest/v1/sql_query?<sql>: GET	263
/db2whrest/v1/sql_query: POST	264
/db2whrest/v1/sql_query_async?<sql>: GET	265
/db2whrest/v1/sql_query_async: POST	266
/db2whrest/v1/task_status/<task_id>: GET and <task_id>/peek	267
/db2whrest/v1/summary_tables -X GET	268
/db2whrest/v1/summary_tables/<table> -X GET	269
/db2whrest/v1/summary_tables/<table> -X PUT	270
/db2whrest/v1/summary_tables/<table>/<action> -X PUT	270
/db2whrest/v1/bulk_add_tags/docs: POST	271
Endpoints for working with policy management	272
/policyengine/v1/policies: GET and /policyengine/v1/policies/<policy_name>: GET	272
/policyengine/v1/policies/<policy_name>/preview: GET	274
/policyengine/v1/policies/<policy_name>/status:GET	275
/policyengine/v1/policyhistory: GET	275
/policyengine/v1/policyhistory/<pol_id>/<log_id>: GET	277
/policyengine/v1/policies/<policy_name>/<action>: POST	278
/policyengine/v1/policies -d '<data>': POST	278
/policyengine/v1/policies/<policy_name>-d '<data>': PUT	279
/policyengine/v1/policies/<policy_name>: DELETE	280
/policyengine/v1/policyhistory: DELETE	281
/policyengine/v1/tags: GET	281
/policyengine/v1/tags/: POST	282
/policyengine/v1/tags/: PUT	283
/policyengine/v1/tags/: DELETE	283
/policyengine/v1/regex: POST	284
/policyengine/v1/regex: GET	284
/policyengine/v1/regex/<regex_name>: GET	286
/policyengine/v1/regex: PUT	286
/policyengine/v1/regex: DELETE	287
Adding fields with the AUTOTAG action	287
Adding fields that contain initial values	288
Adding fields that are extracted by rule from existing fields	288

Endpoints for working with connection management	289
/connmgr/v1/connections: GET and /connmgr/v1/connections/<connection_name>: GET	290
/connmgr/v1/scan/<connection>/partitions: GET	291
/connmgr/v1/connections -d '<data>': POST	291
/connmanager/v1/scan/<connection>/partial: POST	293
/connmgr/v1/connections/<connection_name> -d '<data>': PUT	294
/connmgr/v1/connections/<connection_name>: DELETE	295
/connmgr/v1/scan/<connection_name>: POST	295
/connmgr/v1/scan/<connection_name>: GET	296
/connmgr/v1/scan: GET	297
/connmgr/v1/scan/<connection_name>: PUT	298
Application management by using APIs	299
/policyengine/v1/applications: POST	299
Example: Create a DeepInspect policy using the application	300
/policyengine/v1/applications/<application name>: GET	301
/policyengine/v1/tlscert: GET	302
/policyengine/v1/action_ids: GET	303
/policyengine/v1/applications/<deployment_name>/schema?action_id=<action_id>: GET	304
/policyengine/v1/application_names?action_id=<action_id>: GET	305
/policyengine/v1/applications: DELETE	305
/api/application/appcatalog/publicregistry: GET	306
/api/application/appcatalog/helm: GET	307
/api/application/appcatalog/helm: POST	308
/api/application/appcatalog/helm: PATCH	309
/api/application/appcatalog/helm: DELETE	310
RBAC management by using APIs	310
Managing tokens	311
/auth/v1/token: GET	311
Managing users	312
/auth/v1/users: GET	312
/auth/v1/users: POST	314
/auth/v1/users/<user_ID>/password: POST	314
/auth/v1/users/<user ID>: GET	315
/auth/v1/users/<user_ID>: PATCH	315
/auth/v1/users/<user_ID>: DELETE	316
/auth/v1/users/<user_ID>/groups: GET	316
/auth/v1/users/<user_ID>/collections: GET	317
/auth/v1/users/<user_ID>/roles: GET	318
/auth/v1/users/<user_ID>/role_assignment: GET	319
/auth/v1/users/users_summary: GET	319
Managing user roles	320
/auth/v1/roles/<role_ID>: PUT	320
/auth/v1/roles/<role_ID>: GET	322
/auth/v1/roles: GET	322
/auth/v1/roles/<role_ID>: DELETE	323
Managing user groups	324
/auth/v1/groups: GET	324
/auth/v1/groups: POST	325
/auth/v1/groups/<group ID>: GET	326
/auth/v1/groups/<group_ID>: PATCH	326
/auth/v1/groups/<group_ID>: DELETE	327
/auth/v1/groups/<group_ID>/collections: GET	327
/auth/v1/groups/<group_ID>/roles: GET	328
/auth/v1/groups/<group_id>/role_assignments: GET	328
/auth/v1/groups/<group_ID>/users: GET	329
/auth/v1/groups/<group_ID>/user/<user_ID>: DELETE	330
/auth/v1/groups/<group_ID>/user/<user_ID>: PUT	330
/auth/v1/groups/groups_summary: GET	331
Managing collections	332
/auth/v1/collections: GET	332
/auth/v1/collections: POST	333
/auth/v1/collections/<collection_ID>: GET	333
/auth/v1/collections/<collection_ID>: PATCH	334
/auth/v1/collections/<collection_ID>: DELETE	335
/auth/v1/collections/<collection_ID>/groups: GET	335
/auth/v1/collections/<collection_ID>/users: GET	336
/auth/v1/collections/<collection_id>/role_assignments: GET	336
Managing domains	337
/auth/v1/domains/<domain_ID>: GET	337
/auth/v1/domains: POST	338
/auth/v1/domains/<domain_ID>: PATCH	339
/auth/v1/domains/<domain_ID>: DELETE	339

Graceful shutdown	340
Flushing Kafka topics	342
Returning Data Cataloging to a running state	342
Health check monitoring	344
Multiple connection managers	347
Collecting logs and metrics	347
Creating a Data Cataloging application for metadata-based policies	348
Data Cataloging Harvester CLI	349
FKEY migration script	350
Configurable Db2 log trimmer	351
Disaster recovery	351
Workloads	351
IBM Cloud Paks support for IBM Storage Fusion	352
Validation tools for IBM Cloud Paks	352
IBM Maximo® Applications support for IBM Storage Fusion	353
Knowing your IBM Storage Fusion user interface	354
Browser requirements	355
Icons used in the user interface	355
Serviceability in IBM Storage Fusion	355
Analyzing events in IBM Storage Fusion	356
Collecting logs in IBM Storage Fusion	356
Log package	358
Benefits of enabling call home	358
Data privacy with call home	359
Enabling Call Home	359
IBM Storage Fusion Data Foundation	360
Release notes	360
About this release	361
New features	361
Enhancements	362
Technology Previews	363
Bug fixes	364
Known issues	366
Introduction to Fusion Data Foundation	369
Fusion Data Foundation architecture	370
An overview of Fusion Data Foundation architecture	370
Fusion Data Foundation Operators	371
Fusion Data Foundation operator	371
Components	372
Design diagram	372
Responsibilities	372
Resources	372
Limitation	372
High availability	372
Relevant config files	372
Relevant log files	373
Lifecycle	373
Container Storage operator	373
Components	373
Design diagram	373
Responsibilities	374
Resources	374
Limitation	375
High availability	375
Relevant config files	375
Relevant log files	375
Lifecycle	375
Rook-Ceph operator	375
Components	376
Design diagram	376
Responsibilities	377
Resources	378
Lifecycle	378
MCG operator	379
Components	379
Responsibilities and resources	379
High availability	380
Relevant log files	380
Lifecycle	380
Fusion Data Foundation installation overview	380
Installed Operators	381
OCSInitialization CR	381

Storage cluster creation	381
Internal mode storage cluster	382
Cluster creation	382
NooBaa System creation	382
External mode storage cluster	383
Cluster Creation	383
NooBaa System creation	382
Standalone Multicloud Object Gateway storage cluster	384
NooBaa System creation	382
Storage Cluster creation	385
Fusion Data Foundation upgrade overview	385
Upgrade Workflows	385
ClusterServiceVersion Reconciliation	385
Operator Reconciliation	386
Planning Fusion Data Foundation deployment	386
Storage cluster deployment approaches	386
Internal approach	386
External approach	387
Node types	387
Internal storage services	387
External storage services	388
Security considerations	388
FIPS-140-2	388
Proxy environment	388
Data encryption options	388
Cluster-wide encryption	389
Storage class encryption	389
Data encryption in-transit using messenger version 2 protocol of IBM Storage Ceph	389
Encryption in Transit	389
Subscription offerings	390
Cores versus vCPUs and hyperthreading	390
Cores versus vCPUs and simultaneous multithreading (SMT) for IBM Power	390
Splitting cores	390
Shared processor pools for IBM Power	391
Subscription requirements	391
Infrastructure requirements	391
Platform requirements	391
Amazon EC2	391
Bare Metal	392
VMware vSphere	392
Microsoft Azure	392
Google Cloud	392
Red Hat OpenStack Platform [Technology Preview]	392
IBM Power	392
IBM Z and IBM LinuxONE	392
External mode requirements	393
IBM Storage Ceph	393
IBM FlashSystem	393
Resource requirements	393
Resource requirements for IBM Z and IBM LinuxONE infrastructure	394
Minimum deployment resource requirements	394
Compact deployment resource requirements	395
Resource requirements for MCG only deployment	395
Resource requirements for using Network File system	395
Resource requirements for performance profiles	395
Pod placement rules	395
Storage device requirements	396
Dynamic storage devices	396
Local storage devices	396
Capacity planning	396
Network requirements	396
IPv6 support	397
Multi network plug-in (Multus) support	397
Segregating storage traffic using Multus	397
When to use Multus	398
Multus configuration	398
Requirements for Multus configuration	399
Disaster Recovery	399
Metro-DR	399
Disaster Recovery with stretch cluster	400
Regional-DR	400
Disconnected environment	400
Supported and unsupported features for IBM Power and IBM Z infrastructures	401

Deploying Data Foundation in external mode	401
Deploying multiple Fusion Data Foundation storage clusters	
Overview of multiple storage cluster deployments	403
Preparing to deploy multiple Fusion Data Foundation storage clusters	403
Deploying Fusion Data Foundation Internal storage cluster	404
Deploying Data Foundation external storage cluster	404
Verifying external Data Foundation storage cluster deployment	405
Verifying the state of the pods	405
Verifying the Fusion Data Foundation cluster is healthy	405
Verifying the Multicloud Object Gateway is healthy	406
Verifying that the specific storage classes exist	406
Verifying that Ceph cluster is connected	406
Verifying that storage cluster is ready	406
Migrating application workloads	406
Managing and allocating resources	407
Storage classes	407
Creating storage classes and pools	407
Storage class for persistent volume encryption	408
Access configuration for Key Management System (KMS)	408
Configuring access using vaulttokens	408
Configuring access using vaulttenantsa	409
Creating a storage class for persistent volume encryption	410
Overriding Vault connection details using tenant ConfigMap	412
Storage class with single replica	412
Block pools	413
Creating a block pool	413
Updating an existing pool	413
Deleting a pool	414
Configure storage for OpenShift Container Platform services	414
Configuring Image Registry to use Fusion Data Foundation	414
Using Multicloud Object Gateway as OpenShift Image Registry backend storage	415
Configuring monitoring to use Fusion Data Foundation	417
Overprovision level policy control [Technology Preview]	418
Cluster logging for Fusion Data Foundation	419
Configuring persistent storage	420
Configuring cluster logging to use Fusion Data Foundation	420
Creating Multus networks	421
Creating network attachment definitions	422
Backing OpenShift Container Platform applications with Fusion Data Foundation	422
Adding file and object storage to an existing external Fusion Data Foundation cluster	423
How to use dedicated worker nodes for Fusion Data Foundation	425
Anatomy of an Infrastructure node	425
Managing persistent volume claims	425
Configuring application pods to use Fusion Data Foundation	426
Viewing Persistent Volume Claim request status	426
Reviewing Persistent Volume Claim request events	427
Expanding Persistent Volume Claims	427
Dynamic provisioning	428
Dynamic provisioning in Fusion Data Foundation	428
Available dynamic provisioning plug-ins	429
Reclaiming space on target volumes	429
Enabling reclaim space operation using Annotating Persistent Volume Claims	430
Enabling reclaim space operation using ReclaimSpaceJob	430
Enabling reclaim space operation using ReclaimSpaceCronJob	431
Customizing timeouts required for Reclaim Space Operation	431
Volume snapshots	432
Creating volume snapshots	432
Restoring volume snapshots	433
Deleting volume snapshots	433
Volume cloning	434
Creating a clone	434
Managing container storage interface (CSI) component placements	434
Creating exports using NFS	435
Enabling the NFS feature	435
Creating NFS exports	436
Consuming NFS exports in-cluster	436
Consuming NFS exports externally from the cluster	437
Annotating encrypted RBD storage classes	438
Managing hybrid and multicloud resource	438
Accessing the Multicloud Object Gateway with your applications	439
Accessing the Multicloud Object Gateway from the terminal	439
Accessing the Multicloud Object Gateway from the MCG command-line interface	440
Support of Multicloud Object Gateway data bucket APIs	442
Allowing user access to the Multicloud Object Gateway Console	442

Adding storage resources for hybrid or Multicloud	443
Creating a new backing store	443
Overriding the default backing store	444
Adding storage resources for hybrid or Multicloud using the MCG command line interface	445
Creating an AWS-backed backingstore	445
Creating an AWS-STS-backed backingstore	446
Creating an AWS role using a script	446
Installing OpenShift Data Foundation operator in AWS STS OpenShift cluster	447
Creating a new AWS STS backingstore	448
Creating an IBM COS-backed backingstore	448
Creating an Azure-backed backingstore	449
Creating a GCP-backed backingstore	450
Creating a local Persistent Volume-backed backingstore	451
Creating an s3 compatible Multicloud Object Gateway backingstore	453
Creating a new bucket class	454
Editing a bucket class	454
Editing backing stores for bucket class	454
Managing namespace buckets	455
Amazon S3 API endpoints for objects in namespace buckets	456
Adding a namespace bucket using the Multicloud Object Gateway CLI and YAML	456
Adding an AWS S3 namespace bucket using YAML	456
Adding an IBM COS namespace bucket using YAML	458
Adding an AWS S3 namespace bucket using the Multicloud Object Gateway CLI	459
Adding an IBM COS namespace bucket using the Multicloud Object Gateway CLI	460
Adding a namespace bucket using the OpenShift Container Platform user interface	461
Sharing legacy application data with cloud native application using S3 protocol	462
Creating a NamespaceStore to use a file system	462
Creating accounts with NamespaceStore file system configuration	463
Accessing legacy application data from the openshift-storage namespace	464
Changing the default SELinux label on the legacy application project to match the one in the openshift-storage project	468
Modifying the SELinux label only for the deployment config that has the pod which mounts the legacy application PVC	469
Securing Multicloud Object Gateway	470
Changing the default account credentials to ensure better security in the Multicloud Object Gateway	470
Resetting the noobaa account password	471
Regenerating the S3 credentials for the accounts	471
Regenerating the S3 credentials for the OBC	472
Enabling secured mode deployment for Multicloud Object Gateway	473
Mirroring data for hybrid and Multicloud buckets	474
Creating bucket classes to mirror data using the MCG command-line-interface	474
Creating bucket classes to mirror data using a YAML	474
Bucket policies in the Multicloud Object Gateway	475
Introduction to bucket policies	475
Using bucket policies in Multicloud Object Gateway	475
Creating a user in the Multicloud Object Gateway	476
Multicloud Object Gateway bucket replication	476
Replicating a bucket to another bucket using the MCG command-line interface	477
Replicating a bucket to another bucket using a YAML	477
Setting a bucket class replication policy using the MCG command-line interface	478
Setting a bucket class replication policy using a YAML	479
Object Bucket Claim	480
Dynamic Object Bucket Claim	480
Creating an Object Bucket Claim using the command line interface	481
Creating an Object Bucket Claim using the OpenShift Web Console	483
Attaching an Object Bucket Claim to a deployment	483
Viewing object buckets using the OpenShift Web Console	484
Deleting Object Bucket Claims	484
Caching policy for object buckets	484
Creating an AWS cache bucket	484
Creating an IBM COS cache bucket	485
Lifecycle bucket configuration in Multicloud Object Gateway	487
Scaling Multicloud Object Gateway performance	487
Automatic scaling of MultiCloud Object Gateway endpoints	487
Scaling the Multicloud Object Gateway with storage nodes	488
Increasing CPU and memory for PV pool resources	488
Accessing the RADOS Object Gateway S3 endpoint	488
Using TLS certificates for applications accessing RGW	488
Replacing nodes	489
Replacing nodes on Fusion Data Foundation using dynamic devices	489
Fusion Data Foundation deployed on AWS	489
Replacing an operational AWS node on user-provisioned infrastructure	490
Replacing an operational AWS node on installer-provisioned infrastructure	491
Replacing a failed AWS node on user-provisioned infrastructure	491
Replacing a failed AWS node on installer-provisioned infrastructure	492

Fusion Data Foundation deployed on VMware	493
Replacing an operational VMware node on user-provisioned infrastructure	493
Replacing an operational VMware node on installer-provisioned infrastructure	494
Replacing a failed VMware node on user-provisioned infrastructure	495
Replacing a failed VMware node on installer-provisioned infrastructure	496
Fusion Data Foundation deployed on Microsoft Azure	496
Replacing operational nodes on Azure installer-provisioned infrastructure	497
Replacing failed nodes on Azure installer-provisioned infrastructure	497
Fusion Data Foundation deployed on Google Cloud Platform	498
Replacing an operational node on Google Cloud Platform installer-provisioned infrastructure	498
Replacing a failed Google Cloud Platform node on installer-provisioned infrastructure	499
Replacing nodes on Fusion Data Foundation using local storage devices	500
Replacing storage nodes on bare metal infrastructure	500
Replacing an operational node on bare metal user-provisioned infrastructure	500
Replacing a failed node on bare metal user-provisioned infrastructure	503
Replacing storage nodes on IBM Z or LinuxONE infrastructure	506
Replacing operational nodes on IBM Z or LinuxONE infrastructure	506
Replacing a failed node on an IBM Z infrastructure user provisioned infrastructure	509
Replacing storage nodes on IBM Power infrastructure	511
Replacing an operational or failed storage node on IBM Power	511
Replacing storage nodes on VMware infrastructure	514
Replacing an operational node on VMware user-provisioned infrastructure	515
Replacing an operational node on VMware installer-provisioned infrastructure	517
Replacing a failed node on VMware user-provisioned infrastructure	520
Replacing a failed node on VMware installer-provisioned infrastructure	523
Replacing devices	526
Dynamically provisioned Fusion Data Foundation deployed on AWS	526
Replacing an operational AWS node on user-provisioned infrastructure	490
Replacing an operational AWS node on installer-provisioned infrastructure	491
Replacing a failed AWS node on user-provisioned infrastructure	491
Replacing a failed AWS node on installer-provisioned infrastructure	492
Dynamically provisioned Fusion Data Foundation deployed on VMware	529
Replacing operational or failed storage devices on VMware infrastructure	530
Dynamically provisioned Fusion Data Foundation deployed on Azure	532
Replacing operational nodes on Azure installer-provisioned infrastructure	497
Replacing failed nodes on Azure installer-provisioned infrastructure	497
Dynamically provisioned Fusion Data Foundation deployed on Google Cloud Platform	534
Dynamically provisioned Fusion Data Foundation deployed using local storage devices	534
Replacing operational or failed storage devices on clusters backed by local storage devices	534
Replacing operational or failed storage devices on IBM Power	537
Replacing operational or failed storage devices on IBM Z or LinuxONE infrastructure	542
Monitoring Fusion Data Foundation	542
Cluster health	542
Verifying Fusion Data Foundation is healthy	543
Storage health levels and cluster state	543
Multicloud storage health	543
Enabling multicloud dashboard on Hub cluster	544
Verifying multicloud storage health on hub cluster	544
Metrics	544
Metrics in the Block and File dashboard	545
Metrics in the Object dashboard	546
Pool metrics	547
Network File System metrics	548
Enabling metadata on RBD and CephFS volumes	548
Verify the metadata for RBD PVC	549
Verify the metadata for RBD clone	549
Verify the metadata for RBD snapshots	550
Verify the metadata for RBD Restore	550
Verify the metadata for CephFS PVC	551
Verify the metadata for CephFS clone	552
Verify the metadata for CephFS volume snapshot	553
Verify the metadata of CephFS Restore	553
Alerts	554
Setting up alerts	554
Troubleshooting	554
Downloading log files and diagnostic information using must-gather	555
Commonly required logs for troubleshooting	556
Adjusting verbosity level of logs	557
Overriding the cluster-wide default node selector for Fusion Data Foundation post deployment	557
Encryption token is deleted or expired	558
Troubleshooting alerts and errors in Fusion Data Foundation	558
Resolving alerts and errors	558
Resolving cluster health issues	562

MON_DISK_LOW	562
Resolving cluster alerts	562
CephClusterCriticallyFull	563
CephClusterErrorState	564
CephClusterNearFull	564
CephClusterReadOnly	564
CephClusterWarningState	565
CephDataRecoveryTakingTooLong	565
CephMdsMissingReplicas	566
CephMgrIsAbsent	566
CephMgrIsMissingReplicas	567
CephMonHighNumberOfLeaderChanges	567
CephMonQuorumAtRisk	568
CephMonQuorumLost	568
CephMonVersionMismatch	569
CephNodeDown	569
CephOSDCriticallyFull	570
CephOSDDiskNotResponding	570
CephOSDDiskUnavailable	571
CephOSDFlapping	571
CephOSDNearFull	572
CephOSDSlowOps	572
CephOSDVersionMismatch	572
CephPGRepairTakingTooLong	573
CephPoolQuotaBytesCriticallyExhausted	573
CephPoolQuotaBytesNearExhaustion	573
PersistentVolumeUsageCritical	573
PersistentVolumeUsageNearFull	574
Resolving NooBaa Bucket Error State	574
Resolving NooBaa Bucket Exceeding Quota State	574
Resolving NooBaa Bucket Capacity or Quota State	575
Recovering pods	575
Recovering from EBS volume detach	575
Enabling debug logs for rook-ceph-operator	575
Disabling debug logs for rook-ceph-operator	576
Troubleshooting unhealthy blocklisted nodes	576
Checking for Local Storage Operator deployments	576
Removing failed or unwanted Ceph Object Storage devices	577
Verifying Ceph cluster is healthy	577
Removing failed or unwanted Ceph OSDs in dynamically provisioned Red Hat OpenShift Data Foundation	577
Removing failed or unwanted Ceph OSDs provisioned using local storage devices	578
Troubleshooting the error cephosd:osd.0 is NOT ok to destroy while removing failed or unwanted Ceph OSDs	579
Troubleshooting and deleting remaining resources during Uninstall	579
Troubleshooting CephFS PVC creation in external mode	580
Restoring the monitor pods in Fusion Data Foundation	581
Restoring the Multicloud Object Gateway	585
Restoring ceph-monitor quorum in Fusion Data Foundation	585
Changing resources for the Fusion Data Foundation components	588
Changing the CPU and memory resources on the rook-ceph pods	588
Tuning the resources for the MCG	588
Disabling Multicloud Object Gateway external service after deploying OpenShift Data Foundation	589
Accessing odf-console with the ovs-multitenant plug-in by manually enabling global pod networking	589
Configuring Data Foundation for Disaster Recovery	589
Metro-DR solution for Fusion Data Foundation	590
Components of Metro-DR solution	591
Metro-DR deployment workflow	592
Requirements for enabling Metro-DR	592
Requirements for deploying IBM Storage Ceph stretch cluster with arbiter	592
Hardware requirements	593
Software requirements	593
Network configuration requirements	593
Deploying IBM Storage Ceph	593
Node pre-deployment steps	594
Cluster bootstrapping and service deployment with Cephadm	595
Configuring IBM Storage Ceph stretch mode	599
Installing Fusion Data Foundation on managed clusters	600
Installing Fusion Data Foundation Multicluster Orchestrator operator	602
Configuring SSL access across clusters	602
Creating Disaster Recovery Policy on Hub cluster	603
Configure DRClusters for fencing automation	604
Add node IP addresses to DRClusters	604
Add fencing annotations to DRClusters	604
Create sample application for testing disaster recovery solution	605

Subscription-based applications	
Creating subscription-based sample application	605
Apply DRPolicy to sample application	606
ApplicationSet-based applications	
Creating ApplicationSet-based applications	606
Apply Data policy to sample ApplicationSet-based application	607
Deleting sample application	607
Subscription-based application failover between managed clusters	607
ApplicationSet-based application failover between managed clusters	608
Relocating Subscription-based application between managed clusters	609
Relocating an ApplicationSet-based application between managed clusters	610
Recovering to a replacement cluster with Metro-DR	611
Hub recovery using Red Hat Advanced Cluster Management [Technology preview]	613
Configuring passive hub cluster	613
Switching to passive hub cluster	614
Regional-DR solution for Fusion Data Foundation	614
Components of Regional-DR solution	615
Regional-DR deployment workflow	616
Requirements for enabling Regional-DR	616
Creating a Fusion Data Foundation cluster on managed clusters	617
Installing Fusion Data Foundation Multicluster Orchestrator operator	602
Configuring SSL access across clusters	602
Creating Disaster Recovery Policy on Hub cluster	619
Create sample application for testing disaster recovery solution	605
Subscription-based applications	
Creating a sample subscription-based application	620
Apply DRPolicy to sample application	621
ApplicationSet-based applications	
Creating ApplicationSet-based applications	622
Apply Data policy to sample ApplicationSet-based application	622
Deleting sample application	607
Subscription-based application failover between managed clusters	623
ApplicationSet-based application failover between managed clusters	624
Relocating Subscription-based application between managed clusters	624
Relocating an ApplicationSet-based application between managed clusters	625
Viewing Recovery Point Objective values for disaster recovery enabled applications	625
Hub recovery using Red Hat Advanced Cluster Management [Technology preview]	626
Configuring passive hub cluster	626
Switching to passive hub cluster	626
Disaster recovery with stretch cluster for Fusion Data Foundation	627
Requirements for enabling stretch cluster	628
Applying topology zone labels to OpenShift Container Platform nodes	628
Installing Local Storage Operator	628
Installing IBM Storage Fusion Data Foundation Operator	629
Creating Fusion Data Foundation cluster	629
Verifying Fusion Data Foundation deployment	631
Verifying the state of the pods	631
Verifying the Fusion Data Foundation cluster is healthy	632
Verifying the Multicloud Object Gateway is healthy	632
Verifying that the specific storage classes exist	632
Installing Zone Aware Sample Application	633
Scaling the application after installation	634
Modifying deployment to be Zone Aware	635
Recovering Fusion Data Foundation stretch cluster	636
Understanding zone failure	636
Recovering zone-aware HA applications with rwx storage	636
Recovering HA applications with rwx storage	636
Recovering applications with RWO storage	637
Recovering StatefulSet pods	637
Monitoring disaster recovery health	638
Enable monitoring for disaster recovery	638
Enabling disaster recovery dashboard on Hub cluster	638
Viewing health status of disaster recovery replication relationships	639
Disaster recovery metrics	639
Disaster recovery alerts	640
Troubleshooting disaster recovery	641
Troubleshooting Metro-DR	641
A StatefulSet application stuck after failover	641
DR policies protect all applications in the same namespace	642
During failback of an application stuck in Relocating state	642
Relocate or failback might be stuck in Initiating state	642
Troubleshooting Regional-DR	642

rbd-mirror daemon health is in warning state	642
volsync-rsync-src pod is in error state as it is unable to resolve the destination hostname	643
Cleanup and data sync for ApplicationSet workloads remain stuck after older primary managed cluster is recovered post failover	643
Troubleshooting 2-site stretch cluster with Arbiter	644
Recovering workload pods stuck in ContainerCreating state post zone recovery	644
Troubleshooting	644
Contacting IBM Support Center	644
How to provide feedback	645
Events and error codes message references in IBM Storage Fusion	645
Install events and error codes	646
BMYIN1501	647
BMYIN1502	647
BMYIN1503	647
BMYIN1504	647
BMYIN1505	648
BMYIN1506	648
BMYIN1507	648
BMYIN1508	648
BMYIN1509	648
BMYIN1510	649
BMYIN1512	649
BMYIN1513	649
BMYIN1514	649
BMYIN2501	649
BMYIN2502	650
BMYIN2503	650
BMYIN3116	650
BMYIN3117	650
BMYIN3119	651
BMYPC5010	651
BMYPC6006	651
BMYPC6007	652
BMYPC6008	653
BMYPC6010	654
BMYPC6013	654
Upgrade events and error codes	655
BMYUP1101	655
BMYUP1102	655
BMYUP1103	656
BMYUP1104	656
BMYUP1111	656
BMYUP1112	656
BMYUP1113	656
BMYUP1114	657
BMYUP1115	657
BMYUP1116	657
BMYUP1117	657
BMYUP1118	657
BMYUP1119	658
BMYUP1120	658
BMYUP1121	658
BMYUP1123	658
BMYUP3101	658
BMYUP3102	659
BMYUP3103	659
BMYUP3104	659
BMYUP3105	659
Global Data Platform events and error codes	660
BMYSS0001	660
BMYSS0002	660
BMYSS0003	661
BMYSS0004	661
BMYSS0005	661
BMYSS0006	661
BMYSS0007	661
BMYSS0009	662
BMYSS0010	662
BMYSS0011	662
BMYSS0012	663
BMYSS0013	663
Data foundation events and error codes	663
BMYSS0100	664

BMYSS0101	664
BMYSS0102	664
BMYSS0103	664
BMYSS0104	665
BMYSS0105	665
BMYSS0106	665
BMYSS0107	665
BMYSS0108	666
BMYSS0109	666
BMYSS0110	666
BMYSS0111	666
BMYSS0112	666
BMYSS0115	667
BMYSS0116	667
BMYSS0117	667
BMYSS0118	667
BMYSS0119	668
BMYSS0120	668
BMYSS0121	668
BMYSS0122	668
BMYSS0125	668
BMYSS0126	669
BMYSS0127	669
Data Cataloging events and error codes	669
BMYDC0001	670
BMYDC0002	670
BMYDC0003	670
BMYDC0004	671
BMYDC0005	671
BMYDC0006	671
BMYDC0100	672
BMYDC0101	672
BMYDC0102	672
BMYDC0103	672
BMYDC0104	673
BMYDC0105	673
BMYDC0106	673
BMYDC0107	673
BMYDC0108	674
BMYDC0109	674
BMYDC0110	674
BMYDC0113	674
BMYDC0114	674
BMYDC0115	675
BMYDC0116	675
BMYDC0117	675
BMYDC0118	676
Backup & Restore events and error codes	676
BMYBR0001	676
BMYBR0002	676
BMYBR0003	676
BMYBR0009	677
BMYDP1146	677
BMYDP1163	677
Connection services events and error codes	677
BMYCS100	678
BMYCS101	678
BMYCS102	678
BMYCS103	678
BMYCS300	678
BMYCS301	679
BMYCS302	679
BMYCS303	679
BMYCS304	679
BMYCS305	680
BMYCS306	680
Serviceability events and error codes	680
BMYLC0001	680
BMYLC0002	680
Troubleshooting issues in IBM Storage Fusion	681
Installation and upgrade issues	681
IBM Storage Fusion user interface issues	682

Troubleshooting installation and upgrade issues in IBM Storage Fusion services	682
Global Data Platform service issues	682
Backup & Restore service install and upgrade issues	685
Data Cataloging service issues	688
Common service installation issues	691
Troubleshooting Backup & Restore service issues	692
Backup issues	692
Restore issues	694
Backup & Restore service install and upgrade issues	685
IBM Cloud Pak for Data backup and restore issues	699
Hub and spoke connection issues	700
Backup & restore service from OpenShift Container Platform	701
Troubleshooting Global Data Platform issues	701
Troubleshooting IBM Fusion Data Foundation service	702
IBM Fusion Data Foundation service error scenarios	702
Red Hat OpenShift Data Foundation storage node failure	703
Red Hat OpenShift Data Foundation Object Storage Device (OSD) failure	706
Known issues and limitations	711
IBM Storage Fusion Data Cataloging known issues	711
Frequently asked questions	715
Install	716
Data Foundation Storage	716
Backup and Restore	716
Data Cataloging	717
Support and serviceability	717
Accessibility features for IBM Storage Fusion	719
IBM Storage Fusion considerations for GDPR readiness	720
Glossary	721
A	722
B	722
C	722
D	723
E	724
F	724
G	725
H	725
I	725
K	725
L	726
M	726
N	726
O	726
P	726
R	727
S	727
T	728
U	728
V	729
W	729
Z	729
Notices	729

IBM Storage Fusion documentation

Welcome to the IBM Storage Fusion documentation, where you can find information about how to install, maintain, and use IBM Storage Fusion. This information applies to 2.8 and subsequent 2.8.0 fix pack releases.

Through its integration of a fully containerized version of IBM®'s general parallel file system and backup and restore software, IBM Storage Fusion provides organizations a streamlined way to discover data from across the enterprise. In addition, you can use the software to virtualize and accelerate existing data sets more easily by leveraging the most pertinent storage tier.

With the IBM Storage Fusion solutions, organizations can manage only a single copy of data. No longer are they required to create duplicate data when moving application workloads across the enterprise, easing management functions while streamlining analytics and AI. In addition, data compliance activities (for example, GDPR) can be strengthened by a single copy of data, while security exposure from the presence of multiple copies is reduced.

In addition to its global availability capabilities, IBM Storage Fusion integrates with IBM Cloud® Satellite to help enable businesses to fully manage cloud services at the edge, data center, or in the public cloud with a single management page. IBM Storage Fusion also integrates with Red Hat® Advanced Cluster Management (ACM) for managing multiple Red Hat OpenShift® clusters.

© Copyright IBM Corporation 2024

What's new

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

IBM Storage Fusion 2.8.0 includes new features in the following areas:

- [Backup & Restore](#)
- [Data Cataloging](#)
- [Service upgrades](#)
- [Currency support](#)

Backup & Restore

New recipes for Backup & Restore

Real enterprise applications go well beyond a single 'Deployment' and one persistent volume. They may include operators, span multiple namespaces, and require application data consistency across dozens of persistent volumes. Fusion recipes eliminate the need for multi-page backup run-books by orchestrating backups of even the most complex applications, and doing it all while the application is online. This release includes new recipes for applications, including IBM Cloud Pak for AIOps.

Backup & Restore hardening and serviceability improvements

- Improved security by enabling the use of a read-only root file system by pod containers.
- More detailed error messages for some common missing prerequisites and configuration errors that lead to backup and restore failures.

Backup block-mode PVCs (Technical preview)

IBM Storage Fusion adds support for block-mode persistent volumes. This complements the existing capabilities for backing up file-mode VMs. Now, you can backup OpenShift® VMs regardless of how they access storage. Support for backing up block-mode persistent volumes applies to any application workload; not just VMs. For more information about block-mode PVCs in IBM Storage Fusion, see [Backup & Restore Block PVC Technical preview](#).

Data Cataloging

Harvesters data ingestion for NFS in Data Cataloging

The 2.8 version introduces harvesters as a new strategy for rapid ingestion of NFS distributed data sources and data tagging. This novel way to accelerate the data ingestion on distributed NFS data sources feature allows data cataloging users to take advantage of new entities called harvesters to perform remote data scans and ingestion out of the cluster.

This new capability is useful in the following scenarios:

- The data sources' link speed from the data source to the cluster is low.
- The constraints policies for mounting NFS on the cluster where Data Cataloging service is installed are restrictive

The harvester entity is a software that can be used in a single or multiple distributed computing boxes and ingest data from it.

IBM Db2 upgrade in Data Cataloging

The new IBM Db2 11.5.9 version is supported in Data Cataloging service of IBM Storage Fusion 2.8. This new version provides security fixes, defects correction, improvements to high availability and performance, and other updates. For more information about IBM Db2, see [What's new in Db2 11.5.9](#).

Data Cataloging hardening

- Different Data Cataloging security fixes and defect corrections
- Supports the new AMQ Streams 2.6 version
- Includes log trimmer improvements

Service upgrades

Backup & Restore and Data Cataloging service upgrades just got easier with built in common pre-checks that automatically validate the health of these services. During upgrade, you can easily 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.

Currency support

- **Red Hat® OpenShift Container Platform version**

IBM Storage Fusion HCI System supports OpenShift Container Platform 4.15.

- **IBM Storage Fusion service versions**

- IBM Storage Scale or Global Data Platform service 5.2.0.0
- Fusion Data Foundation 4.15
- Data Cataloging 2.1.6
- Backup & Restore 2.8

◦ For supported IBM Cloud Paks versions, see [IBM Cloud Paks support for IBM Storage Fusion](#).

Hotfixes

The following table lists the hotfixes required for the IBM Storage Fusion 2.8.x release. These updates are necessary for error-free functionality.

Hotfix for 2.8.0

The following table lists the issues fixed by this hotfix.

Table 1. Issues fixed in 2.8.0 hotfix

Issues fixed	Version	Reference links
Global Data Platform OpenShift® Container Platform issues	2.8.0	For more information about this issue, see https://www.ibm.com/support/pages/node/7157065

Product overview

Organizations must quickly adjust to the changing business and outside influences that are causing rapid change, resulting in a need for business agility and faster business insights. Clients need applications and data to adjust and shift in response to dynamic market demands. They also need diverse and simplified tools and data services to build anywhere, at any pace, and for applications and data to scale dynamically, achieve peak performance, and adhere to security requirements.

Companies need a consistent way to deploy applications across on-premises infrastructure and public clouds, and not all of them are deploying containers to create that portability and consistency between cloud and on-premises environments.

IBM Storage 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 Storage 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.

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

IBM Storage Fusion

It is a software-defined storage management software with protection, backup, and caching elements and can be run on existing hardware resources.

IBM Storage Fusion HCI System

It is a purpose-built, hyper-converged architecture that is designed to deploy bare metal Red HatOpenShift container management and deployment software alongside IBM Storage 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 Storage Fusion provides a comprehensive infrastructure with compute, networking, and storage resources, including a data platform and global data services for Red Hat OpenShift.

To know more about IBM Storage Fusion HCI System, see <https://ibm.com/docs/en/sfhs/2.8.x>.

- **[Benefits of IBM Storage Fusion](#)**

Benefits offered by IBM Storage 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 simple straightforward pricing and selection of services.

- **[IBM Storage Fusion trial version](#)**

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

- **[Fusion Essentials](#)**

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

- **[Security in IBM Storage Fusion](#)**

IBM Storage 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 Storage Fusion are signed. This page describes how to verify the signatures on those images.

Benefits of IBM Storage Fusion

Benefits offered by IBM Storage Fusion.

- Modernize AI workloads
 - Supports GPU accelerated applications with NVIDIA A100 GPU nodes and integrates IBM Cloud Paks to infuse AI. For more details about the supported versions of IBM Cloud Paks and prerequisites, see [IBM Cloud Paks support for IBM Storage Fusion](#).
- Organize and optimize resources
 - Get file, volume, and object access across disparate data sources — including traditional storage, IBM storage, as well as 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 disc 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 Storage 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 simple straightforward pricing and selection of services.

For more information about IBM Storage Expert Care for IBM Storage Fusion, see [Support reference guide](#).

IBM Storage Fusion trial version

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

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

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

Fusion Essentials

Fusion Essentials refers to the restricted use of IBM Storage 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 Storage 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 Storage 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 Storage 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 Storage Fusion Operator in your cluster. For more information about the installation, see [Deploying IBM Storage Fusion](#).
2. Deploy and configure the Fusion Data Foundation service in your cluster. For more information about the Fusion Data Foundation installation, see [Data Foundation](#).

Security in IBM Storage Fusion

IBM Storage Fusion is a secure platform to deploy your applications.

The following security measures were followed for IBM Storage 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 Storage 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.

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 Storage 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 Storage 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-----

```
mQINBGPOHO1BEADA253R4yBkEFFO7d9gfu+hdG58qQe2H+4m5MuLc0zy/ulhkGg
YGBWFBCutkMp9F/jvtTrhf8r0oS1rIIzjYHfpM6qn1LftkbzQlpaj8yT+aPpNgVm
1xF5+hS32ouPX1ufaP3sX5CL0VZ4Uh10FBszfps7EPQU4C9fp6Rwstgejt+k5xc
L1GiU1a+Kr3bmrBqN8nhsTE5cm1zS6tAIiG1R1I3qGRVcICpPh31N7Mw2PYH8mb
kr+0pWQlJa+V9HxDx0LCY8aPuxfLZUbTKEVkuGMXWvBHPVNYScydos5Jd9
Lzpc5uP6a8+7SBdmGZ5rupVkdKAZRTn+bvx149D1yO0DBbXLaOMYevXUCIDeq3DI
X5BLNUzy8j6283IEReBTol8HnefpCT4mXQijFCIk595JiZwbcCY23XddooXetwk1
WZQj+TeQdV2+gSS+tmdj31D/i0lXbNkneBK1G858sgct2hTiqZ/s2B3R941DL2N
neV41csRy4r00bG0dyIL/24yV857U427s1x56zc740pYgF/Ju61dauvixyaUUD
C2KBy/ZWmpWaMAecNLRRYcHCUB9HZCfWchPlwyXWIlggjL3D8vRpPePWITpb1c4wF
Tl6utwKHKM1CUqjn20j6q2Mh/7Z7g5XwqamtDm8cpxlRY2ChHHxPxzbQARAQAB
tCpJQk0gU3B1Y3RydW0gRnVzaW9uIEhDSSA8cHNpcnRAxDMuaWJtLmNvbT6JAjoE
EwEIAcQFAmPOH0ICGw8FCwkIBwIGFQoJCAsCBYCAwEChgEFCQPCzv8ACgkQHS4H
jVltgQNUmQ/+J9xUtUs1PwzAB/LpoqHGOMyZqz6EmQbaRiPzD4VSxcTYbpSKVb07
5H1sV80gDR90UT5cbc4d9NbtfG1Y8UeiCSL+MPjy5ksjFmbsxLRKEYZjB/qIOHv
c8viMH4y0boquomjeqGL1qajKyUvG1h99S5osGJVmpa2tmxMhTa157+aDX5SAKS4
R01fd52K4nZw5+icq95fXoEkPNwd6Xm2fZ10F+kZ2dS8y+gu45hEaa/h+3P2myJ
37mp+336cfqrqmqlvOpXD64et8zsQRNYqr2QEUSNs7ok4RkRqlpkv5RYTV7T7qH
g88rLAY1vxw4RzkQMve+t1IqgLANG9A+/fbQUQ/yshjCmvqmzB54VrOjOiwMWVtL
J+u1jbrdX1kFPogei6c0v3CBNe1khO6+1amEPtazKONbuMhBvaseX16x4Mk5+0FS
4FLNDEjktk4qUQAKEZk/Sugopz6oQpMYmKxPiT/7snLL9e6naF98TosKrpUckme3
phbRIStsML7Z7rUWq+jGSO1z2L4FnVgrFs8iKsq23jvAf02nTKptfgbozp6WMRC/KL
Wn39bGePGCMwPNwn3Au08w/qMvHNAHkwTq1T1r/1vrmonx+EEar2J2ZxqoA82P
4hcXWG4auoSA3y9tjAVLWChZdawFcMqcHZ1HxER9/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 Storage Fusion from enterprise registry](#).

In this procedure, the following example image is used: `icr.io/cpopen/isf-operator-catalog:2.8.0-linux.amd64`.

Procedure

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

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

To confirm the key deletion, enter 'Y'. If you have previous IBM Storage 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 Storage Fusion public key on the machine that you prepared according to the *Before you begin* section:

```
sudo qpq2 --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 qpg2 --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  
skopeo copy docker://icr.io/cpopen/isf-operator-catalog:2.7.1-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).

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 iamanikey:<YOUR_ENTITLEMENT_KEY>`

- #### 5. Verify the signature:

Deploying IBM Storage Fusion

Prerequisites and procedure to install IBM Storage Fusion on OpenShift® Container Platform that runs on On-premises VMware, On-premises Bare Metal, Linux on IBM zSystems, 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 Storage Fusion.
 - **[Installing IBM Storage Fusion on On-premises VMware](#)**
Procedure to install IBM Storage Fusion on OpenShift Container Platform that runs on On-premises VMware.
 - **[Installing IBM Storage Fusion on On-premises Bare Metal](#)**
Procedure to install IBM Storage Fusion on OpenShift Container Platform that runs on On-premises Bare Metal.
 - **[Installing IBM Storage Fusion on On-premises Linux on IBM zSystems and zCX](#)**
Procedure to install IBM Storage Fusion on OpenShift Container Platform that runs on On-premises Linux on IBM zSystems and zCX.
 - **[Installing IBM Storage Fusion on IBM Cloud](#)**
Procedure to install IBM Storage Fusion on OpenShift Container Platform that runs on IBM Cloud.
 - **[Installing IBM Storage Fusion on Amazon Web Services](#)**
Procedure to install IBM Storage Fusion on OpenShift Container Platform that runs on Amazon Web Services.
 - **[Installing IBM Storage Fusion on Microsoft Azure](#)**
Procedure to install IBM Storage Fusion on OpenShift Container Platform that runs on Microsoft Azure.
 - **[Installing IBM Storage Fusion on On-premises IBM Power Systems](#)**
Procedure to install IBM Storage Fusion on OpenShift Container Platform that runs on On-premises IBM Power Systems.
 - **[Installing IBM Storage Fusion on Google Cloud](#)**
Procedure to install IBM Storage Fusion on OpenShift Container Platform that runs on Google Cloud.
 - **[Installing IBM Storage Fusion from enterprise registry](#)**
Prerequisites and procedure to install IBM Storage Fusion on OpenShift Container Platform.
 - **[Validating IBM Storage Fusion installation](#)**
Steps to confirm the success of the installation.
 - **[Uninstalling IBM Storage Fusion](#)**
Steps to uninstall IBM Storage Fusion from Red Hat® OpenShift web console.

Prerequisites

Prerequisite that you need for a successful setup and installation of IBM Storage 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 Storage Fusion services, go through their respective *Before you begin* section. For information about the services, see [Managing services](#).
 3. Ensure that you have the infrastructure and Red Hat® OpenShift® Container Platform 4.12, 4.14, or 4.15.
- Note:
- If you want to install Data Foundation 4.14 or 4.15 storage in consumer mode, then OpenShift Container Platform must also be the same version.
 - You can upgrade Red Hat OpenShift Container Platform to the supported 4.15 version. However, you cannot enable Data Cataloging service in an upgraded IBM Storage Fusion 2.8 with Red Hat OpenShift Container Platform 4.15.
- For the supported version of Red Hat OpenShift Container Platform, see <https://www.ibm.com/support/pages/node/7081324>.
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 Storage Fusion on IBM cloud, see [Creating image pull secret for IBM Cloud based installation](#).
6. If you plan to deploy IBM Storage 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.

- [Obtaining entitlement key](#)
Entitlement keys determine whether IBM Storage 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 Storage 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.
- [System requirements](#)
The system requirements for installing IBM Storage 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 Storage 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 Storage 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

- Run the following command to log in to OpenShift system:

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

- 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](#).

- 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.

- 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**.

- 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
```

- 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.

- 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

- To add private registries, edit the global **pull-secret** in the **openshift-config** project as follows:

- 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 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
```

System requirements

The system requirements for installing IBM Storage Fusion software.

The IBM Storage Fusion operators and your workloads run on OpenShift® Container Platform that in turn are installed on amd64: 64-bit Intel, AMD x86, or Linux on IBM zSystems hardware architecture. The following table lists the system requirements for IBM Storage 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".

IBM Fusion Data Foundation can be deployed on Infra or worker nodes, whereas all other IBM Storage Fusion services including Base must be deployed on worker nodes.

Component	vCPUs	Memory	Storage
IBM Storage Fusion Base	Request CPU: 4 Limit CPU: 10	Request memory: 4 GiB Limit memory: 13 GiB	Overall storage is a minimum of 500 MB.

Component	vCPUs	Memory	Storage
IBM Fusion Data Foundation 4.15.x	Internal mode (Local and dynamic) Request CPU: 22+2*(total count of Object Storage Daemon) Limit CPU: 22+2*(total count of Object Storage Daemon) External mode Request CPU: 4 Limit CPU: 5 For IBM Power Systems: Internal mode 48 CPU (logical) External mode 24 CPU (logical)	Internal mode (Local and dynamic) Request memory: 56+5*(total count of Object Storage Daemon) GiB Limit memory: 56+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 IBM 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
Global Data Platform	Per node Request CPU: 10% (total vCPU of the node) Cluster Request CPU: 3 Limit CPU: 12	Per node Request memory: 10% (total memory of the node) Cluster Request memory: 13 Limit memory: 30	For more information about IBM Spectrum Scale Container Native Storage Access, see Hardware requirements for IBM Storage Scale Container Native Storage Access .
Backup & Restore hub	Request CPU: 5 Limit CPU: 30	Request memory: 17 GiB Limit memory: 41 GiB	200 GB Minimum
Backup & Restore spoke	Request CPU: 2 Limit CPU: 8	Request Memory: 2 GiB Limit Memory 7 GiB	10GiB
Data Cataloging	Request CPU: 14 Limit CPU: 77	Request memory: 29 GiB Limit memory: 162 GiB	500 GB Minimum For more information about Data Cataloging, see Data Cataloging .

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	4.12, 4.14, 4.15 Note: You can upgrade Red Hat OpenShift Container Platform to the supported 4.15 version. However, you cannot enable Data Cataloging service in an upgraded IBM Storage Fusion 2.8.0 with Red Hat OpenShift Container Platform 4.15.
Storage	IBM Fusion Data Foundation 4.12, 4.14, and 4.15 Note: For deployment platforms and supported services, see IBM Storage Fusion Services support matrix . IBM Storage Scale Remote mount of IBM Storage Scale storage: Note: IBM Storage Fusion 2.8.0 or higher supports IBM Storage Scale 5.2.0.0. <ul style="list-style-type: none"> • IBM Storage Scale remote storage cluster release 5.1.3 or higher. • The IBM Storage 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 33.00. For more information, see Software requirements . 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 .

Component	Versions
Backup & Restore	<p>IBM Storage Fusion supports backup and recovery operations on Red Hat OpenShift Container Platform environment on AMD64 with any storage provider that implemented the Container Storage Interface (CSI) driver and meets the following prerequisites:</p> <ul style="list-style-type: none"> The Container Storage Interface driver must be v1 or higher (no support for <i>alpha</i> or <i>beta</i> versions) The Container Storage Interface driver must support Volume Snapshot and Restore. StorageClass must be created with <code>volumeBindingMode: Immediate</code> (<code>volumeBindingMode: WaitForFirstConsumer</code> is not supported) and <code>allowVolumeExpansion: true</code>. PersistentVolume (PV) must be created with <code>volumeMode: Filesystem</code> (<code>volumeMode: Block</code> is not supported) <p>Note: It is also recommended that the <code>VolumeSnapshotClass</code> deletion policy is configured to delete snapshots after the corresponding <code>VolumeSnapshotContent</code> is deleted (<code>deletionPolicy: Delete</code>).</p> <p>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.</p>

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.openshift.com/container-platform/4.15/authentication/managing-security-context-constraints.html>.

The IBM Storage Scale specific constraints are as follows:

Note: IBM Storage Fusion 2.8.0 or higher supports IBM Storage Scale 5.2.0.

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 Storage Fusion installation

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

In IBM Storage 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.

Check the following table for the image size details for each operator:

Table 1. Image size

Operator	Image size
Red Hat OpenShift® Container Platform	50 GB - 60 GB
IBM Storage Fusion	10 GB
Data Foundation	62 GB
Global Data Platform	11 GB
Backup & Restore	30 GB
Data Cataloging	64 GB

Mirroring your images to the enterprise registry

If you are planning a disconnected or offline installation, then you must mirror images to your enterprise registry.

Before you begin

- The registry must have at least one directory path specified.

For example:

```
https://<enterprise registry host>:<enterprise registry port>/<mandatory root path>
```

- Considerations for your enterprise registry:**

- There must not be a huge latency between cluster nodes and your enterprise registry.
- You must have a container image registry that supports Docker v2-2 in the location that hosts the Red Hat® OpenShift® Container Platform cluster. For more information about image manifest, see [opm CLI reference](#).

Important: Artifactory is the recommended registry because it supports the following must have capabilities:

- Import and export
- Untagged images

JFrog Artifactory version 7.55.8 is tested on IBM Storage Fusion. The port that is specified in the URL is used to login and pull the images from the enterprise repository. For a secured enterprise registry, specify 443. If you do not provide the port value, then no default port is considered in its absence. The API key is the only supported authentication method.

- Ensure that your secure enterprise registry is already setup and ready for use.

- Prepare your mirroring host:

- The mirror host must have access to internet and enterprise registry.
- Ensure that you install the following tools on your system from where you can connect to Red Hat registry and enterprise registry:
 - Install Docker or Podman.
 - Install `opm` CLI tool. For more information about installing `opm` CLI tool, see [Installing the opm CLI](#).
 - Install skopeo 1.14 or higher for image copy operation. For more information to install skopeo, see <https://github.com/containers/skopeo>.
 - Install `oc` command tool from any of the Red Hat OpenShift Container Platform cluster:
 - Log in to the system from where you want to run commands.
 - Log in to OpenShift Container Platform.
 - Click bell icon and select Command line tools.
 - 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/clients/ocp/4.10.21/>

For example, use the following link to download `oc` client for Linux platform: <https://mirror.openshift.com/pub/openshift-v4/clients/ocp/4.10.21/openshift-client-linux-4.10.21.tar.gz>

- Download `pull-secret.txt`.

To download `pull-secret`, see <https://console.redhat.com/openshift/install/pull-secret> and follow the instructions.

Edit the downloaded `pull-secret` with registry credentials:

Add a new section of key-value pair under `auths`. For example,

```
"<Your enterprise registry>:<port>": {
  "auth": "<base64 encoded 'user_name:password'>",
  "email": "<your email>" }
```

See the following sample values:

```
{
  "auths": {
    "cloud.openshift.com": {
      ...
      "registryhost.com:443": {
        "auth": "dXNlc19uYW1lOnBhc3N3b3Jk",
        "email": "user_name@ibm.com"
      }
    }
  }
}
```

Here, `user_name` and `password` are credentials to connect to enterprise registry.

Note: If you want to use multiple repositories, add auth section for both repositories.

- Ensure that you have entitlement key to access IBM Storage Fusion images.

- Note:

- Ensure that the `LOCAL_ISF_REPOSITORY` path and `TARGET_PATH` of the IBM Storage Fusion and IBM Storage Scale images must be same.
- IBM Storage Fusion 2.8.0 or higher supports IBM Storage Scale 5.2.0.

For example, if you use `LOCAL_ISF_REPOSITORY=sds/mirror-2.8.0` while mirroring IBM Storage Fusion images, the same path must be used for IBM Storage Scale.

About this task

Points to note about this task:

- Repository and target path must be same for IBM Storage 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 Storage Fusion services you want to enable you can choose to only mirror images for that service, however you must mirror IBM Storage Fusion images.

Procedure

1. Run the following steps to mirror IBM Storage Fusion images.
For the actual procedure, see [Mirroring IBM Storage Fusion images](#).
2. Mirror IBM Storage Scale images.
For the actual procedure, see [Mirroring IBM Storage Scale images](#).
3. Mirror Data Foundation images deployed on OpenShift Container Platform version 4.12.
For the actual procedure, see [Mirroring Data Foundation images deployed on OpenShift Container Platform version 4.14 or 4.15](#).
4. Mirror Backup & Restore images.
For the actual procedure, see [Mirroring Backup & Restore images](#).
5. Mirror Data Cataloging images.
For the actual procedure, see [Mirroring Data Cataloging images](#).

Mirroring Red Hat operator images to enterprise registry

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

Before you begin

- Make sure that you go through the *Before you begin* section and *About the task* section of [Mirroring your images to the enterprise registry](#). For more information about the installation of Red Hat operator images, see [Using Operator Lifecycle Manager on restricted networks](#).
- Make sure 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](#).
- The registry must have at least one directory path specified.
For example:

```
https://<enterprise registry host>:<enterprise registry port>/<mandatory root path>
```

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:

- If you face any issues during mirroring Red Hat operator 4.15 images, then use either Ubuntu 22 or higher or RHEL 9 or higher server versions to mirror. For more information, see the Red Hat issue <https://access.redhat.com/solutions/7062641>.
- 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-2.8.0/isf` or `sds-2.8.0`.

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 Storage Fusion installation.

For example:

```
cat << EOF > imageset-config-redhatoperator.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
mirror:
  operators:
    - catalog: registry.redhat.io/redhat/redhat-operator-index:$OCP_VERSION
      packages:
        - name: "amq-streams"
        - name: "redhat-oadp-operator"
EOF
```

Note:

- 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 `amq-streams` package is required to install the Data Cataloging service. If you plan to use the IBM Data Cataloging service, retain it in the commands, or else you can skip it.

- Run the following `oc` command to mirror the images from the specified image set configuration to a specified registry.

```
oc mirror --config imageset-config-redhatoperator.yaml docker://$TARGET_PATH --dest-skip-tls --ignore-history
```

This can take 5-10 minutes to complete.

Example output:

```
Rendering catalog image "<TARGET_PATH>/redhat/redhat-operator-index:v4.15" 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
```

- After you mirror the content to your registry, go to the generated `oc-mirror-workspace/` directory. Go to the `results-1xxxx` directory, and verify that the YAML files are present for the `ImageContentSourcePolicy` and `CatalogSource` resources.

- Apply the generated `ImageContentSourcePolicy` to the cluster.

```
cd ./oc-mirror-workspace/results-<generated_id>
oc apply -f imageContentSourcePolicy.yaml
```

You can run this step only once when the cluster is up for a freshly installed rack.

- Apply the generated `CatalogSource` to the cluster.

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

Mirroring IBM Storage Fusion images

Mirror the IBM Storage Fusion images to your enterprise registry.

About this task

If you do not have Docker, you can use Podman.

Procedure

- 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>
```

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 Storage 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>
IFS='/' read -r NAMESPACE PREFIX <<< "$LOCAL_ISF_REPOSITORY"
if [[ "$PREFIX" != "" ]]; then export TARGET_PATH="$LOCAL_ISF_REGISTRY/$NAMESPACE/$PREFIX"; export REPO_PREFIX=$(echo
"$PREFIX" | sed -r 's/^\//-/g'); else export TARGET_PATH="$LOCAL_ISF_REGISTRY/$NAMESPACE"; export REPO_PREFIX=""; fi
#verify both variables set correctly
echo "$TARGET_PATH"
echo "$REPO_PREFIX"
```

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-2.8.0/isf` or `sds-2.8.0`.

- 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>
```

- From the mirroring host, run the following copy command to copy IBM Storage Fusion images to the host:

Note:

- Make sure that you are logged in to the source and destination repositories by using the docker login command.
- If you mirror to a Quay registry, use a destination tag name (`<image-name>:xxxxxx`) rather than the digest name to prevent Red Hat Quay from deleting your downloaded images.

```
skopeo copy --all docker://cp.icr.io/cp/isf-sds/<image-name>@sha256:xxxxxx docker://$TARGET_PATH/<image-name>:xxxxxx
```

Example skopeo command:

```
skopeo copy --all docker://cp.icr.io/cp/isf-sds/fusion-
ui@sha256:705b20557b63315e6ddc403808aa57ee9cf0a621d531ddf2f3f196b63ca11acb docker://$TARGET_PATH/fusion-
ui:705b20557b63315e6ddc403808aa57ee9cf0a621d531ddf2f3f196b63ca11acb
```

For more information about Red Hat Quay garbage collection, see https://access.redhat.com/documentation/en-us/red_hat_quay/3.10/html/manage_red_hat_quay/garbage-collection.

```
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-
sds/callhomeclient@sha256:964e5d16b2a570d15a54effbb73f653298012fbe53a8532f83a228a3af68a8e7
docker://$TARGET_PATH/callhomeclient@sha256:964e5d16b2a570d15a54effbb73f653298012fbe53a8532f83a228a3af68a8e7
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-
sds/eventmanager@sha256:1586572baef1fc4cd5bc7e68f3f37cd8d5c3e2c788b6d3d23f14c2e5854d448
docker://$TARGET_PATH/eventmanager@sha256:1586572baef1fc4cd5bc7e68f3f37cd8d5c3e2c788b6d3d23f14c2e5854d448
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/eventmanager-
snmp@sha256:f379b7df7fcee22ce51a34464455309dc803786d523cb6b8c40a2bddf73341f95 docker://$TARGET_PATH/eventmanager-
snmp@sha256:f379b7df7fcee22ce51a34464455309dc803786d523cb6b8c40a2bddf73341f95
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/fusion-
ui@sha256:705b20557b63315e6ddc403808aa57ee9cf0a621d531ddf2f3f196b63ca11acb docker://$TARGET_PATH/fusion-
ui@sha256:705b20557b63315e6ddc403808aa57ee9cf0a621d531ddf2f3f196b63ca11acb
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/isf-application-
operator@sha256:8b654f88051f81746d3cb0a350c38f30df5da3151a4b8846655ca4fdc6238788 docker://$TARGET_PATH/isf-application-
operator@sha256:8b654f88051f81746d3cb0a350c38f30df5da3151a4b8846655ca4fdc6238788
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/isf-cns-
operator@sha256:13217921c6c64607ad0cd3cbc14fd156af3ec5f439b96ec9f81670dc574a53e docker://$TARGET_PATH/isf-cns-
operator@sha256:13217921c6c64607ad0cd3cbc14fd156af3ec5f439b96ec9f81670dc574a53e
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/isf-data-protection-
operator@sha256:204a23a5836d1274481576b46681bd80ae06240ace2740036010f6448650fe11 docker://$TARGET_PATH/isf-data-
protection-operator@sha256:204a23a5836d1274481576b46681bd80ae06240ace2740036010f6448650fe11
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/isf-management-
consoleplugin@sha256:ddbab64895bea43b0fa1fc3ccb39d3ac2a12f23b041db70793696d3822bb3153 docker://$TARGET_PATH/isf-
management-consoleplugin@sha256:ddbab64895bea43b0fa1fc3ccb39d3ac2a12f23b041db70793696d3822bb3153
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/isf-prereq-
operator@sha256:05b169dcf76ad9a7b34eb85abf441c0836318cec14072a86a5f4a00e01803a86 docker://$TARGET_PATH/isf-prereq-
operator@sha256:05b169dcf76ad9a7b34eb85abf441c0836318cec14072a86a5f4a00e01803a86
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/isf-
proxy@sha256:e7ace0060bd473743e70cdfe1c469394600c2b6642bd9a23e8858d3bea53da35 docker://$TARGET_PATH/isf-
proxy@sha256:e7ace0060bd473743e70cdfe1c469394600c2b6642bd9a23e8858d3bea53da35
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/isf-serviceability-
operator@sha256:f8f4460b9de754ff1f58c027596e8dee0a621d008b1bb52746263be127863a docker://$TARGET_PATH/isf-serviceability-
operator@sha256:f8f4460b9de754ff1f58c027596e8dee0a621d008b1bb52746263be127863a
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-sds/isf-ui-
operator@sha256:a40346c44a49e7990e75fab813f2ef209272043a57c7a6499d8ce0d805bc9cde docker://$TARGET_PATH/isf-ui-
operator@sha256:a40346c44a49e7990e75fab813f2ef209272043a57c7a6499d8ce0d805bc9cde
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/isf-
sds/logcollector@sha256:9238c5cfe576e0511849bdb3bf75f39f11fbc71310ee4db49d783717190243f2
docker://$TARGET_PATH/logcollector@sha256:9238c5cfe576e0511849bdb3bf75f39f11fbc71310ee4db49d783717190243f2
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/isf-operator-software-
bundle@sha256:b79a0d3ee86abb12636b6cdc02486da3fef09584b7c953be7ee765bfa516c21 docker://$TARGET_PATH/isf-operator-
software-bundle@sha256:b79a0d3ee86abb12636b6cdc02486da3fef09584b7c953be7ee765bfa516c21
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/isf-operator-software-catalog:2.8.0
docker://$TARGET_PATH/isf-operator-software-catalog:2.8.0
skopeo copy --insecure-policy --preserve-digests --all docker://registry.redhat.io/openshift4/ose-kube-rbac-
proxy@sha256:767682dd3bd65651a8204c9fb732b9c48b99127189992b43abc6eccce027f4589 docker://$TARGET_PATH/openshift4/ose-kube-
rbac-proxy@sha256:767682dd3bd65651a8204c9fb732b9c48b99127189992b43abc6eccce027f4589
```

Note: Ensure all the copy commands complete successfully without errors. For example, if you have the correct entitlement key but still observe the following error for any or all of the copy commands, then contact [IBM support](#):

denied: insufficient scope

4. Add `ImageContentSourcePolicy`.

Note:

- The `ImageContentSourcePolicy` contains list of repositories, copy digests for the IBM Storage Fusion.
- Replace the variable `$TARGET_PATH` with your registry details where images are mirrored.

See the following `ImageContentSourcePolicy`:

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: isf-fusion-icsp
spec:
  repositoryDigestMirrors:
    - mirrors:
        - $TARGET_PATH
      source: cp.icr.io/cp/isf-sds
    - mirrors:
        - $TARGET_PATH
      source: icr.io/cpopen
    - mirrors:
        - $TARGET_PATH/openshift4
      source: registry.redhat.io/openshift4
```

Mirroring IBM Storage Scale images

Mirror the IBM Storage Scale images to your enterprise registry.

About this task

For more reference information, see [Offline setup for network restricted Red Hat OpenShift Container Platform clusters in IBM Spectrum Scale Container Native 5.2.0](#).

Note:

- This procedure is only necessary if you plan to enable the Global data platform IBM Storage Fusion service.
- IBM Storage Fusion 2.8.0 or higher supports IBM Storage Scale 5.2.0.

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 Storage 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-2.8.0/isf or sds-2.8.0.

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: Make sure you are logged into the source and destination repositories via docker login command.

```
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/data-management/ibm-spectrum-scale-
daemon@sha256:65a1b65e4076ac7be840904a6a1d59eb74849cd58a1a4a59b2f9a8de94652de docker://$TARGET_PATH/data-management/ibm-
spectrum-scale-daemon@sha256:65a1b65e4076ac7be840904a6a1d59eb74849cd58a1a4a59b2f9a8de94652de
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-core-
init@sha256:089718b2d96d909b86bf19448c4d86f31a3877eb450d465d830e55133415015d docker://$TARGET_PATH/ibm-spectrum-scale-
core-init@sha256:089718b2d96d909b86bf19448c4d86f31a3877eb450d465d830e55133415015d
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
gui@sha256:93288c812f0bace075eb785f519d67f684bd33407c44a93b53df9abd16e0a docker://$TARGET_PATH/ibm-spectrum-scale-
gui@sha256:93288c812f0bace075eb785f519d67f684bd33407c44a93b53df9abd16e0a
skopeo copy --all --preserve-digests
docker://cp.icr.io/cp/spectrum/scale/postgres@sha256:bdd7346fab25b7e0b25f214829d6ebfb78ef0465059492e46dee740ce8fc844
docker://$TARGET_PATH/postgres@sha256:bdd7346fab25b7e0b25f214829d6ebfb78ef0465059492e46dee740ce8fc844
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/ubi-
minimal@sha256:bc552efb4966aaa44b02532b3e168acf18e2af9d0fe89502a1d9fabafbc5 docker://$TARGET_PATH/ubi-
minimal@sha256:bc552efb4966aaa44b02532b3e168acf18e2af9d0fe89502a1d9fabafbc5
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
pmcollector@sha256:07a9e951ab315724ca39e332a7559ca98519442b0669241fd97ffdb3a0381667 docker://$TARGET_PATH/ibm-spectrum-
scale-pmcollector@sha256:07a9e951ab315724ca39e332a7559ca98519442b0669241fd97ffdb3a0381667
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
monitor@sha256:3f41d20e7beaf757778d8650172adf8456569607dab3479eb4524407ed2e4a13 docker://$TARGET_PATH/ibm-spectrum-scale-
monitor@sha256:3f41d20e7beaf757778d8650172adf8456569607dab3479eb4524407ed2e4a13
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-grafana-
bridge@sha256:a6ca689d8205f17bb278910c521842f98ced8537aba55b287792a91269bf2f41 docker://$TARGET_PATH/ibm-spectrum-scale-
grafana-bridge@sha256:a6ca689d8205f17bb278910c521842f98ced8537aba55b287792a91269bf2f41
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
coredns@sha256:88cbfe40fd302a6467cb7e852b298f6c8d8659782ab313706d491d3ddf172a6e docker://$TARGET_PATH/ibm-spectrum-scale-
coredns@sha256:88cbfe40fd302a6467cb7e852b298f6c8d8659782ab313706d491d3ddf172a6e
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/csi/csi-
snapshotter@sha256:1a29ab1e4ecd33a84062cec757620d9787c28b28793202c5b78ae097c3dee27 docker://$TARGET_PATH/csi/csi-
snapshotter@sha256:1a29ab1e4ecd33a84062cec757620d9787c28b28793202c5b78ae097c3dee27
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/csi/csi-
attacher@sha256:d69cc72025f7c40dae112ff989e920a3331583497c8dfb1600c5ae0e37184a29 docker://$TARGET_PATH/csi/csi-
attacher@sha256:d69cc72025f7c40dae112ff989e920a3331583497c8dfb1600c5ae0e37184a29
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/csi/csi-
provisioner@sha256:de79c8bbc271622eb94d2ee8689f189ea7c1cb6adac260a421980fe5eed66708 docker://$TARGET_PATH/csi/csi-
provisioner@sha256:de79c8bbc271622eb94d2ee8689f189ea7c1cb6adac260a421980fe5eed66708
skopeo copy --all --preserve-digests
docker://cp.icr.io/cp/spectrum/scale/csi/livenessprobe@sha256:5baeb4a6d7d517434292758928bb33efc6397368ccb48c8a4cf29496abf4
e987 docker://$TARGET_PATH/csi/livenessprobe@sha256:5baeb4a6d7d517434292758928bb33efc6397368ccb48c8a4cf29496abf4e987
```

```

skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/csi/csi-node-driver-
registrar@sha256:c53535af8a7f7e3164609838c4b191b42b2d81238d75c1b2a2b582ada62a9780 docker://$TARGET_PATH/csi/csi-node-
driver-registrar@sha256:c53535af8a7f7e3164609838c4b191b42b2d81238d75c1b2a2b582ada62a9780
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale/csi/csi-
resizer@sha256:4c148bbdf883153bc72d321be4dc55c33774a6d98b2b3e0c2da6ae389149a9b7 docker://$TARGET_PATH/csi/csi-
resizer@sha256:4c148bbdf883153bc72d321be4dc55c33774a6d98b2b3e0c2da6ae389149a9b7
skopeo copy --all --preserve-digests docker://cp.icr.io/cp/spectrum/scale-csi-
driver@sha256:b2bc343eadbc11d9ed74a8477d2cd0a7a8460a72203d3f6236d4662e68df1166 docker://$TARGET_PATH/csi/ibm-spectrum-
scale-csi-driver@sha256:b2bc343eadbc11d9ed74a8477d2cd0a7a8460a72203d3f6236d4662e68df1166
skopeo copy --all --preserve-digests docker://icr.io/cpopen/ibm-spectrum-scale-
operator@sha256:d44f11cc61f410dd20d7bb52ac28eebc80f928eede9be434c270a7ad5648b626 docker://$TARGET_PATH/ibm-spectrum-scale-
operator@sha256:d44f11cc61f410dd20d7bb52ac28eebc80f928eede9be434c270a7ad5648b626
skopeo copy --all --preserve-digests docker://icr.io/cpopen/ibm-spectrum-csi-
operator@sha256:bd264199ac10d574163bfa32bb88844fd786ee6f794a56e235591d2f051c7807 docker://$TARGET_PATH/ibm-spectrum-scale-
csi-operator@sha256:bd264199ac10d574163bfa32bb88844fd786ee6f794a56e235591d2f051c7807
skopeo copy --all --preserve-digests docker://icr.io/cpopen/ibm-spectrum-scale-must-
gather@sha256:eb5b2e72579b96f2fc04162e4dd6ace7eb3570f2f08dca552c045ea29faa3d docker://$TARGET_PATH/ibm-spectrum-scale-
must-gather@sha256:eb5b2e72579b96f2fc04162e4dd6ace7eb3570f2f08dca552c045ea29faa3d

```

- In the OpenShift® Container Platform cluster, add `ImageContentSourcePolicy`:

Note:

- The `ImageContentSourcePolicy` contains list of repositories, copy digests for the IBM Storage Scale.
- Replace the variable `$TARGET_PATH` with your registry details where images are mirrored.

See the following `ImageContentSourcePolicy`.

```

apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: isf-scale-icsp
spec:
  repositoryDigestMirrors:
    # for scale
    - mirrors:
      - $TARGET_PATH
      source: cp.icr.io/cp/spectrum/scale
    - mirrors:
      - $TARGET_PATH
      source: icr.io/cpopen

```

Do not add white spaces while you update the mirror path in `ImageContentSourcePolicies`. 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

```

Mirroring Data Foundation images deployed on OpenShift Container Platform version 4.14 or 4.15

Mirror the Data Foundation images to your enterprise registry.

Before you begin

- Ensure that you use OpenShift® Container Platform 4.14.4 or higher in the commands. Use your specific version number in the commands.
- Make sure OpenShift Container Platform version is at least 4.14.4 or higher.

About this task

The version 4.15 is used in the procedure as an example. Please replace with your appropriate supported version.

For more information about how to plan for this setup in a network restricted Data Foundation, see [Disconnected environment](#).

Note: This procedure is only necessary if you plan to enable the IBM Fusion Data Foundation service.

Procedure

- 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>
```

- 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 Storage 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 4.14>"
```

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, sds-2.8.0/isf or sds-2.8.0.

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 IBM Fusion Data Foundation.

See the following image set configuration:

```
cat << EOF > imageset-config-fdf.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: "$TARGET_PATH/isf-df-metadata:latest"
    skipTLS: true
  mirror:
    operators:
      - catalog: icr.io/cpopen/isf-data-foundation-catalog:v<OCP_VERSION>
      packages:
        - name: "mcg-operator"
        - name: "ocs-operator"
        - name: "odf-csi-addons-operator"
        - name: "odf-multicloud-orchestrator"
        - name: "odf-operator"
        - name: "odr-cluster-operator"
        - name: "odr-hub-operator"
        - name: "ocs-client-operator"
EOF
```

Run the following oc command to mirror the images:

```
oc mirror --config imageset-config-fdf.yaml docker://{$TARGET_PATH} --dest-skip-tls --ignore-history
```

5. Create the image set configuration for local storage operator.

See the following image set configuration:

```
cat << EOF > imageset-config-lso.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: "$TARGET_PATH/df/odf-lso-metadata:latest"
    skipTLS: true
  mirror:
    operators:
      - catalog: registry.redhat.io/redhat/redhat-operator-index:v<OCP_VERSION>
      packages:
        - name: "local-storage-operator"
        - name: "lvms-operator"
EOF
```

Run the following oc command to mirror the images:

```
oc mirror --config imageset-config-lso.yaml docker://{$TARGET_PATH} --dest-skip-tls --ignore-history
```

Note: If Red Hat® operator catalog is running on your cluster, prune and push all earlier packages, including the `local-storage-operator` and `lvms-operator`. Otherwise, old packages get lost from the Red Hat operator index image.

6. Add `ImageDigestMirrorSet` for IBM Fusion Data Foundation.

Note: The `ImageDigestMirrorSet` contains a list of repositories, copy digests for the IBM Fusion Data Foundation.

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-idsp
spec:
  imageDigestMirrors:
    - mirrors:
        - <enterprise registry host:port>/<target-path>/openshift4
      source: registry.redhat.io/openshift4
    - mirrors:
```

```

- <enterprise registry host:port>/<target-path>/redhat
  source: registry.redhat.io/redhat
- mirrors:
  - <enterprise registry host:port>/<target-path>/rhel8
    source: registry.redhat.io/rhel8
- mirrors:
  - <enterprise registry host:port>/<target-path>/cp/df
    source: cp.icr.io/cp/df
- mirrors:
  - <enterprise registry host:port>/<target-path>/cpopen
    source: cp.icr.io/cpopen
- mirrors:
  - <enterprise registry host:port>/<target-path>/cpopen
    source: icr.io/cpopen
- mirrors:
  - <enterprise registry host:port>/<target-path>/cp/ibm-ceph
    source: cp.icr.io/cp/ibm-ceph
- mirrors:
  - <enterprise registry host:port>/<target-path>/lvms4
    source: registry.redhat.io/lvms4

```

7. Create a CatalogSource named `redhat-operators`.

Note: Do this step when the 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: <enterprise registry host:port>/<target-path>/redhat/redhat-operator-index:v<OCP_VERSION>
  publisher: Red Hat
  sourceType: grpc

```

8. Add `ImageTagMirrorSet` for Data Foundation:

```

apiVersion: config.openshift.io/v1
kind: ImageTagMirrorSet
metadata:
  name: isf-fdf
spec:
  imageTagMirrors:
    - mirrors:
      - <enterprise registry host:port>/<target-path>/df/isf-data-foundation-catalog
        source: icr.io/cpopen/isf-data-foundation-catalog
  mirrorSourcePolicy:
    AllowContactingSource 0

```

Migrating your Data Foundation ImageContentSourcePolicy to ImageDigestMirrorSet

Migrate the Fusion Data Foundation ImageContentSourcePolicy to an ImageDigestMirrorSet.

About this task

Using an `ImageContentSourcePolicy` (ICSP) object to configure repository mirroring is a deprecated feature. Though this function is still included in Red Hat® OpenShift® Container Platform, do not use it for new deployments.

On Red Hat OpenShift Container Platform version 4.14.4, the `ImageContentSourcePolicy` can coexist with the `ImageDigestMirrorSet`. You can migrate the Fusion Data Foundation `ImageContentSourcePolicy` to an `ImageDigestMirrorSet`.

Procedure

- Log in to the Red Hat OpenShift Container Platform as a user with sufficient permissions to complete the task.

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

- Run the following command to get the name of the image content source policies on your cluster.

```
oc get ImageContentSourcePolicy
```

- Set the `FDF_ICSP` environment variable to the name of the Fusion Data Foundation `ImageContentSourcePolicy`.

The default name of the Fusion Data Foundation `ImageContentSourcePolicy` is `isf-fdf-icsp`. The following command uses the default name:

```
export FDF_ICSP=isf-fdf-icsp
```

- Save the `ImageContentSourcePolicy` as a YAML file on the cluster.

The following command saves the YAML file to the current directory:

```
oc get ImageContentSourcePolicy ${FDF_ICSP} -o yaml >> ${FDF_ICSP}.yaml
```

- Convert the `ImageContentSourcePolicy` to an `ImageDigestMirrorSet`:

```
oc create -f $(oc adm migrate icsp ${FDF_ICSP}.yaml | cut -f 4 -d ' ')
```

Note: This command might trigger node upgrades. Wait for all the nodes to be in Ready state before you proceed to the next step.
6. Delete the ImageContentSourcePolicy:

```
oc delete ImageContentSourcePolicy ${FDF_ICSP}
```

Note: This command might trigger node upgrades. Wait for all the nodes to be in Ready state before you proceed to the next step.
7. If ImageTagMirrorSet for Data Foundation does not exist, then add it.

```
apiVersion: config.openshift.io/v1
kind: ImageTagMirrorSet
metadata:
  name: isf-fdf
spec:
  imageTagMirrors:
    - mirrors:
      - <enterprise registry host:port>/<target-path>/df/isf-data-foundation-catalog
        source: icr.io/cpopen/isf-data-foundation-catalog
  mirrorSourcePolicy:
    AllowContactingSource 0
```

8. Optional: Remove the image digest value in fusionservicedefinition/data-foundation-service.

Before you remove the image digest, confirm that the current digest value is the same as the current tag name. Use **skopeo** to get the digest ID of your current tag.
a. Query the imageDigest from your enterprise registry. Replace v4.14 with your current Red Hat OpenShift Container Platform version.

```
skopeo inspect docker://<enterprise registry host:port>/<target-path>/cpopen/isf-data-foundation-catalog:v4.14 | jq -r ".Digest"
```

b. Run the following **OC** command.

```
oc edit fusionservicedefinition data-foundation-service -n ibm-spectrum-fusion-ns
```

Here, replace **ibm-spectrum-fusion-ns** with your Fusion namespace.

Warning: The imageDigest value from [8.a](#) must be consistent with **fsd data-foundation-service** CR spec:

```
.spec.onboarding.multiVersionCatSrcDetails.ocp414-t`
```

if the two digestIDs are different, IBM Storage Fusion triggers a Data Foundation upgrade. Make sure the tag-based catalog image (**<enterprise registry host:port>/<target-path>/cpopen/isf-data-foundation-catalog:v4.14**) is always the latest.

c. If you confirm the same digest ID or accept the upgrade, you can delete the line as follows and save it.

Example:

```
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
      ocp414-t:
        displayName: Data Foundation Catalog
        imageDigest: sha256:2d9e78d69a457b722cf6037968dae5f48eccd9e48ef4369a7fe661de0d96df95  <-- delete this line
        imageName: isf-data-foundation-catalog
        imageTag: v4.14
        publisher: IBM
        registryPath: icr.io/cpopen
        skipCatSrcCreation: false
```

d. Verify whether the catalog source image is updated.

Example:

```
# oc get catalogsources.operators.coreos.com -n openshift-marketplace isf-data-foundation-catalog -o jsonpath='{.spec.image}'
icr.io/cpopen/isf-data-foundation-catalog:v4.14
```

Mirroring Backup & Restore images

Mirror the Backup & Restore images to your enterprise registry.

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 Backup & Restore IBM Storage Fusion service.

Procedure

- 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>
```

Run the following command to login to Red Hat **quay.io** with your Red Hat enterprise credentials:

```
docker login quay.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 Storage 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"
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.

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-2.8.0/isf or sds-2.8.0.

- 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>
```

- 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 [3](#).

- 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.

- From the mirroring host, run the following copy command to copy images to be mirrored for Backup & Restore:

Note: Make sure that you are logged in to the source and destination repositories through the docker login command.

```
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/fbr/guardian-application-service@sha256:4af659712ed613d110383f4ac43a102ba887795763f312a8bf8db835653b5cf9 docker://$TARGET_PATH/guardian-application-service@sha256:4af659712ed613d110383f4ac43a102ba887795763f312a8bf8db835653b5cf9
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/fbr/guardian-backup-location@sha256:62658a1735311606d584857f902f2c4a423f1c3e85b2ea289ae54727193cd825 docker://$TARGET_PATH/guardian-backup-location@sha256:62658a1735311606d584857f902f2c4a423f1c3e85b2ea289ae54727193cd825
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/fbr/guardian-backup-policy@sha256:89e21c7425ea72aef2f3f8e53c36a74d8beccb561d3b1e3476b75e22004dd6d49 docker://$TARGET_PATH/guardian-backup-policy@sha256:89e21c7425ea72aef2f3f8e53c36a74d8beccb561d3b1e3476b75e22004dd6d49
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/fbr/guardian-backup-service@sha256:56e4119fc3a89fe8a1c2babffaaabf3d91aeecc8076023347a5381c607d4ae3 docker://$TARGET_PATH/guardian-backup-service@sha256:56e4119fc3a89fe8a1c2babffaaabf3d91aeecc8076023347a5381c607d4ae3
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/fbr/guardian-job-manager@sha256:d5eb66db95e62170874e2d0f3ec5e097819596768f90f650013a2456a0fc83b7 docker://$TARGET_PATH/guardian-job-manager@sha256:d5eb66db95e62170874e2d0f3ec5e097819596768f90f650013a2456a0fc83b7
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/fbr/guardian-transaction-manager@sha256:f09efb569de67917d41a6b3f374dceee4703f859787513ca452a2c5ad965d3f58 docker://$TARGET_PATH/guardian-transaction-manager@sha256:f09efb569de67917d41a6b3f374dceee4703f859787513ca452a2c5ad965d3f58
skopeo copy --insecure-policy --preserve-digests --all docker://cp.icr.io/cp/fbr/guardian-kubevirt-velero-plugin@sha256:f96d83593f5053c1ce1619243c88ff7fa4a5e3a6a1f9b4df944b45d73211f3e docker://$TARGET_PATH/guardian-kubevirt-velero-plugin@sha256:f96d83593f5053c1ce1619243c88ff7fa4a5e3a6a1f9b4df944b45d73211f3e
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/guardian-datamover@sha256:76c3f168faec35d1cf7c56010646b731123ba123abb1a60427409edd13ec6b70 docker://$TARGET_PATH/guardian-datamover@sha256:76c3f168faec35d1cf7c56010646b731123ba123abb1a60427409edd13ec6b70
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/guardian-dm-operator@sha256:35562a8b290b600c626d81b1e573931d33f2e15e2b0a9d46cd3fb59e23ce0ae8 docker://$TARGET_PATH/guardian-dm-operator@sha256:35562a8b290b600c626d81b1e573931d33f2e15e2b0a9d46cd3fb59e23ce0ae8
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/guardian-dp-operator@sha256:faec1b7a85aab714716a850ac4109db0079e6214f3e06b1d8a91cd5b73fa5f docker://$TARGET_PATH/guardian-dp-operator@sha256:faec1b7a85aab714716a850ac4109db0079e6214f3e06b1d8a91cd5b73fa5f
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/idp-agent-operator@sha256:3a927d363fccfc3ca75023d2c94ed923e6e86828a7d086d790e999b0f0e1ff1 docker://$TARGET_PATH/idp-agent-
```

```

operator@sha256:3a927d363fccfc3ca75023d2c94ed923e6e86828a7d086d790e999b0f0e1ff1
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/idp-server-
operator@sha256:50f119e4ae22cc21a4b9b7fe5d3b1da4b8352c2d4635b58bbe596aae62b36d22 docker://$TARGET_PATH/idp-server-
operator@sha256:50f119e4ae22cc21a4b9b7fe5d3b1da4b8352c2d4635b58bbe596aae62b36d22
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/idp-server-operator-
bundle@sha256:aadfc6b18c33ed632f660ec5df9e1bf7ed866359737f61ffe27bc308e9816567 docker://$TARGET_PATH/idp-server-operator-
bundle@sha256:aadfc6b18c33ed632f660ec5df9e1bf7ed866359737f61ffe27bc308e9816567
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/idp-server-operator-
catalog@sha256:5462246b796dd0c6e729c8802491b7cd5bb59cb98476ab118bee6a1f520c5c2d docker://$TARGET_PATH/idp-server-operator-
catalog@sha256:5462246b796dd0c6e729c8802491b7cd5bb59cb98476ab118bee6a1f520c5c2d
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/guardian-mongo-
operator@sha256:006765a2a3a5451b8eebdc8e0ec10a4384124e99f7099f281ab9f04b13d9e357 docker://$TARGET_PATH/guardian-mongo-
operator@sha256:006765a2a3a5451b8eebdc8e0ec10a4384124e99f7099f281ab9f04b13d9e357
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/guardian-redis-
operator@sha256:cb698df0f89b5f3c04191dd1902dc1e29269544e7720987aa557fb6b6130d1le0e docker://$TARGET_PATH/guardian-redis-
operator@sha256:cb698df0f89b5f3c04191dd1902dc1e29269544e7720987aa557fb6b6130d1le0e
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/guardian-dm-operator-
bundle@sha256:938433010f4ed3b71e7eaba41020c5114604ce0d77300aa3efbf2c5fe5465095 docker://$TARGET_PATH/guardian-dm-operator-
bundle@sha256:938433010f4ed3b71e7eaba41020c5114604ce0d77300aa3efbf2c5fe5465095
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/guardian-dp-operator-
bundle@sha256:a8fbec8721edb951d02566369df74eea161ad859b3e28d9e44ce7b642fb104d1 docker://$TARGET_PATH/guardian-dp-operator-
bundle@sha256:a8fbec8721edb951d02566369df74eea161ad859b3e28d9e44ce7b642fb104d1
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/idp-agent-operator-
bundle@sha256:a4528cc6d4f63e64326dd519fbec19625933f61e150ad3a6f7ba9bcd5a9dcb2 docker://$TARGET_PATH/idp-agent-operator-
bundle@sha256:a4528cc6d4f63e64326dd519fbec19625933f61e150ad3a6f7ba9bcd5a9dcb2
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/guardian-mongo-operator-
bundle@sha256:79a86e9c116046304fe03f54d8f87b0f7b6e6293a2900d437e8138ab1ff7481 docker://$TARGET_PATH/guardian-mongo-
operator-bundle@sha256:79a86e9c1160463045fe03f54d8f87b0f7b6e6293a2900d437e8138ab1ff7481
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/guardian-redis-operator-
bundle@sha256:ec8bd43483d77e6b0b41daf77d8dbbd83e54fee233e3a7faa714cbf6dalb3f6 docker://$TARGET_PATH/guardian-redis-
operator-bundle@sha256:ec8bd43483d77e6b0b41daf77d8dbbd83e54fee233e3a7faa714cbf6dalb3f6
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/fbr-mongo-
exporter@sha256:21aeeb051009d7784be0e0a8e86440dca26ac2ef57ffc461611de0414f1bb676 docker://$TARGET_PATH/fbr-mongo-
exporter@sha256:21aeeb051009d7784be0e0a8e86440dca26ac2ef57ffc461611de0414f1bb676
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/fbr-
mongo@sha256:73a3b08cd819a2175ae3308dc72fa07518d0417071f79196338f636219127f4 docker://$TARGET_PATH/fbr-
mongo@sha256:73a3b08cd819a2175ae3308dc72fa07518d0417071f79196338f636219127f4
skopeo copy --insecure-policy --preserve-digests --all docker://icr.io/cpopen/fbr-
redis@sha256:2131345868484fe9540746c9e4a42a9b71dbbb80de5916d74604f2bfc4ec0f5 docker://$TARGET_PATH/fbr-
redis@sha256:2131345868484fe9540746c9e4a42a9b71dbbb80de5916d74604f2bfc4ec0f5
skopeo copy --insecure-policy --preserve-digests --all docker://quay.io/openshift-release-dev/ocp-v4.0-art-
dev@sha256:a731786db7f7bc7a4137309a16a7d27acbb9fd21c5731602e2ccaadc58155805 docker://$TARGET_PATH/ocp-v4.0-art-
dev@sha256:a731786db7f7bc7a4137309a16a7d27acbb9fd21c5731602e2ccaadc58155805
skopeo copy --insecure-policy --preserve-digests --all docker://quay.io/openshift-release-dev/ocp-v4.0-art-
dev@sha256:bf7bbc0987b7514ab15f3e20f1b432ee19d09f83c80729008b718e158f78a5f6 docker://$TARGET_PATH/ocp-v4.0-art-
dev@sha256:bf7bbc0987b7514ab15f3e20f1b432ee19d09f83c80729008b718e158f78a5f6
skopeo copy --insecure-policy --preserve-digests --all
docker://quay.io/minio/minio@sha256:a9cec9ed5bda5b4e1b2153823e01b309965b9de7ed6eb7f098d45592eecdfc78
docker://$TARGET_PATH/minio@sha256:a9cec9ed5bda5b4e1b2153823e01b309965b9de7ed6eb7f098d45592eecdfc78
skopeo copy --insecure-policy --preserve-digests --all docker://registry.redhat.io/rhel19/redis-
7@sha256:33d868f8832be3dbe73adaft7e394646c8eda91af9fb79d5e800c755925878c docker://$TARGET_PATH/rhel19/redis-
7@sha256:33d868f8832be3dbe73adaft7e394646c8eda91af9fb79d5e800c755925878c

```

Ensure all commands are successful.

5. Add `ImageContentSourcePolicy` after `skopeo` commands are executed successfully on the cluster.

See the following `ImageContentSourcePolicy`:

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

```

apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  labels:
    operators.openshift.org/catalog: "true"
  name: ifbr-offline-mirrors
spec:
  repositoryDigestMirrors:
    - mirrors:
        - $TARGET_PATH
        source: cp.icr.io/cp/fbr
    - mirrors:
        - $TARGET_PATH
        source: icr.io/cpopen
    - mirrors:
        - $TARGET_PATH
        source: quay.io/minio

```

Mirroring Data Cataloging images

Mirror the Data Cataloging images to your enterprise registry.

Before you begin

- Make sure 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 plug-in](#).
- Make sure that you use OpenShift® Container Platform version 4.12 or higher in the commands. Use your specific version number in the commands.
- Make sure OpenShift Container Platform version is at least 4.12.
- Install either docker or podman to run login commands.

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 Data Cataloging IBM Storage Fusion HCI System service.
- The destination registry where all 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.

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: Make sure that your entitlement key for IBM Storage 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.

```
export LOCAL_ISF_REGISTRY=<Your enterprise registry host>
```

`LOCAL_ISF_REGISTRY` is your enterprise registry. Examples based on IBM Registry: `cp.icr.io`, `cp.icr.io:443` and `cp.icr.io/my/custom/path`.

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 Data Cataloging.

```
cat << EOF > imagesetconfiguration_dcs.yaml
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: "${LOCAL_ISF_REGISTRY}/isf-dcs-metadata:latest"
    skipTLS: true
  mirror:
    operators:
      - catalog: "oci:///tmp/dcs_catalog"
        packages:
          - name: "ibm-spectrum-discover-operator"
      - catalog: icr.io/cpopen/ibm-operator-catalog:latest
        packages:
          - name: "db2u-operator"
        channels:
          - name: "v110509.0"
    additionalImages:
      - name: icr.io/db2u/db2u@sha256:793e05f77076e8e1e055c738e90c9706d582c7da286e908955504842f844ce06
      - name: icr.io/db2u/db2u.restricted@sha256:afd250032aeaf39549f43d992a487a9d7b004f074e2cce3d8aadf6a8f327cecfb
      - name: icr.io/db2u/db2u.db.wh@sha256:0886ee44e500e571ffcb47a82701b616da94fe9ff9eb2c0582a0852c30194fa
      - name: icr.io/db2u/db2u.dv.api@sha256:621f1226c0041ec8961a219e804abb6b45a0b9332a9dc2dcdb216549035ee9b5
      - name: icr.io/db2u/db2u.dv.caching@sha256:86be7fd023978bfaa922da408d457d864a0ad353d18cd07770f28a5b924fe41b
      - name: icr.io/db2u/db2u.dv.utils@sha256:79258a4b20e4063109e943c7f0aeab513bb837ddcd7dfb2ce72e9d1d598b2bb9
      - name: icr.io/db2u/etc@sha256:d1dd2eae940427ff7bcd40506cc2181f3fc7826d48dbbb3a4bc3349a2d8c2f93
      - name: icr.io/db2u/db2u.logstreaming.fluentd@sha256:55fbdb2c1938cb0f735e32a8385748b702c38c3e41c83cb44e8b3d5e41b685269
      - name: icr.io/db2u/db2u.hurricane@sha256:c28133ed6c25c163b5c63957773f7ff3375874f7a40bec9d3579bfc1349a78eb
      - name: icr.io/db2u/db2u.instdb@sha256:cf2f99358fb6beac6ca2f9553855b8f33d4cfcd7748bd191c66b725180722cab
      - name: icr.io/db2u/db2u.instdb.restricted@sha256:a78c2c6f7b43e857554e1eed21a9d1fd41202559cb9e4231d975c48c0eb7075b
      - name: icr.io/db2u/db2u.auxiliary.auth@sha256:90d80d10fa6573ea466512a3fc88c9f80ccb67f4d188206fa515b15993c56a96
      - name: icr.io/db2u/db2u.mustgather@sha256:a8a5f6ab563a7fdb93f22ee2eda3a2250c297c67011a1aabdb46d6c68482535a3
      - name: icr.io/db2u/db2u.qrep@sha256:c1121abab93bee6cf1d27019a2ba4ac2b5f7ade980207713054b184812d6d5
      - name: icr.io/db2u/db2u.rest@sha256:7cda76f07c2d6ced608befb3a4cf6f88c331bb8b3a54d02e6b3c6c03f8abf92d2
      - name: icr.io/db2u/db2u.update.image@sha256:68cac0eb685747fdbd88cb160ea859298bd3a86c3d31a2ddd6b2da28180e56fc
      - name: icr.io/db2u/db2u.tools@sha256:290532cb23d45a246dad7bca1aa761407480a3423f645800f5aa6caldedd863d
      - name: icr.io/db2u/db2u.veleroplugin@sha256:01369d5f1da84f5ceb170c9bb9287e4fdac1dd39f30515b6a13ae37c1f38c559
```

```

- name: icr.io/db2u/db2u.watsonquery@sha256:e1c9542fd5a6ea90b7a265d823e348aaba669545f817f47ce7076dc5753963fe
EOF

6. Prepare the Data Cataloging catalog image and registries configuration.

skopeo --override-os=linux copy docker://icr.io/cpopen/ibm-spectrum-discover-operator-
catalog@sha256:c2538264cb1882b1c98fea5ef162f198ce38ed8c940e82e3b9db458a9a46cb15 oci:///tmp/dcs_catalog --format v2s2
skopeo --override-os=linux copy --all docker://icr.io/cpopen/ibm-spectrum-discover-operator-
catalog@sha256:c2538264cb1882b1c98fea5ef162f198ce38ed8c940e82e3b9db458a9a46cb15 docker://${LOCAL_ISF_REGISTRY}/cpopen/ibm-
spectrum-discover-operator-catalog@sha256:c2538264cb1882b1c98fea5ef162f198ce38ed8c940e82e3b9db458a9a46cb15

cat << EOF > registries_dcs.conf
[[registry]]
location = "icr.io/cp/ibm-spectrum-discover"
insecure = false
blocked = false
mirror-by-digest-only = true
prefix = ""

[[registry.mirror]]
location = "cp.icr.io/cp/ibm-spectrum-discover"
insecure = false
EOF

7. Mirror the images using the image set configuration.

oc mirror --config imagesetconfiguration_dcs.yaml docker://${LOCAL_ISF_REGISTRY} --dest-skip-tls --ignore-history --oci-
registries-config registries_dcs.conf

8. Create an ImageContentSourcePolicy for IBM Storage Fusion Data Cataloging.

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

9. Create the IBM Storage Fusion operators and Red Hat operators catalogs.

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

```

Installing IBM Storage Fusion on On-premises VMware

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

Before you begin

- If you plan to do an offline installation of IBM Storage Fusion, see [Enterprise registry for IBM Storage 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 [IBM Storage Fusion Services support matrix](#).

About this task

Support is available for only one instance of IBM Storage 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 Storage 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 Storage Fusion**.
 5. Click IBM Storage Fusion.
The Version, Capability level, Source, and Provider type of IBM Storage Fusion is available.
 6. Click Install.
It opens the Install Operator page for **IBM Storage 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 Storage 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.
 13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Storage 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 Storage Fusion dialog box gets displayed.
 15. In the IBM Storage 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 Storage Fusion installation](#).
2. If you want to enable the services from the IBM Storage Fusion user interface, see the following procedures accordingly:
 - a. For Data Cataloging, see [Data Cataloging](#).
 - b. For Data Foundation, see [Data Foundation](#).
 - c. For Backup & Restore, see [Backup & Restore](#).

If you want to install IBM Storage Fusion services using OpenShift Container Platform, then see [Deploying services from OpenShift Container Platform](#).
Important: To know more about the supported services for your platform, see [IBM Storage Fusion Services support matrix](#).
3. To know more about the user interface of IBM Storage Fusion, see [Knowing your IBM Storage Fusion user interface](#).

Installing IBM Storage Fusion on On-premises Bare Metal

Procedure to install IBM Storage 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 Storage Fusion, see [Enterprise registry for IBM Storage Fusion installation](#).
- For more information about deployments and their supported services, see [IBM Storage Fusion Services support matrix](#).

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 Storage 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 Storage 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 Storage Fusion**.

5. Click IBM Storage Fusion.
The Version, Capability level, Source, and Provider type of IBM Storage Fusion is available.
 6. Click Install.
It opens the Install Operator page for **IBM Storage 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 Storage 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-operator** 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 Storage 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 Storage Fusion dialog box gets displayed.
 15. In the IBM Storage 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 Storage Fusion installation](#).
 2. If you want to enable the services from the IBM Storage Fusion user interface, see the following procedures accordingly:
 - a. For Data Cataloging, see [Data Cataloging](#).
 - b. For Data Foundation, see [Data Foundation](#).
 - c. For Backup & Restore, see [Backup & Restore](#).
- If you want to install IBM Storage Fusion services using OpenShift Container Platform, then see [Deploying services from OpenShift Container Platform](#).
Important: To know more about the supported services for your platform, see [IBM Storage Fusion Services support matrix](#).
3. To know more about the user interface of IBM Storage Fusion, see [Knowing your IBM Storage Fusion user interface](#).

Installing IBM Storage Fusion on On-premises Linux on IBM zSystems and zCX

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

Before you begin

If you plan to do an offline installation of IBM Storage Fusion, see [Enterprise registry for IBM Storage 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 [IBM Storage Fusion Services support matrix](#).

About this task

Support is available for only one instance of IBM Storage 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 Storage 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 Storage Fusion**.
5. Click IBM Storage Fusion.
The Version, Capability level, Source, and Provider type of IBM Storage Fusion is available.
6. Click Install.
It opens the Install Operator page for **IBM Storage 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 Storage 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.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Storage 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 Storage Fusion dialog box gets displayed.
15. In the IBM Storage 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 Storage Fusion installation](#).
 2. If you want to enable the services from the IBM Storage Fusion user interface, see the following procedures accordingly:
 - a. For Data Cataloging, see [Data Cataloging](#).
 - b. For Data Foundation, see [Data Foundation](#).
 - c. For Backup & Restore, see [Backup & Restore](#).
- If you want to install IBM Storage Fusion services using OpenShift Container Platform, then see [Deploying services from OpenShift Container Platform](#).
Important: To know more about the supported services for your platform, see [IBM Storage Fusion Services support matrix](#).
3. To know more about the user interface of IBM Storage Fusion, see [Knowing your IBM Storage Fusion user interface](#).

Installing IBM Storage Fusion on IBM Cloud

Procedure to install IBM Storage 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 [IBM Storage Fusion Services support matrix](#).

For steps to create a image pull secret for installing IBM Storage 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 Storage 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 Storage 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 Storage Fusion**.
5. Click IBM Storage Fusion.
The Version, Capability level, Source, and Provider type of IBM Storage Fusion is available.
6. Click Install.
It opens the Install Operator page for **IBM Storage 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 Storage 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.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Storage 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 Storage Fusion dialog box gets displayed.
15. In the IBM Storage 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 Storage Fusion installation](#).
 2. If you want to enable the services from the IBM Storage Fusion user interface, see the following procedures accordingly:
 - a. For Data Cataloging, see [Data Cataloging](#).
 - b. For Data Foundation, see [Data Foundation](#).
 - c. For Backup & Restore, see [Backup & Restore](#).
- If you want to install IBM Storage Fusion services using OpenShift Container Platform, then see [Deploying services from OpenShift Container Platform](#).
Important: To know more about the supported services for your platform, see [IBM Storage Fusion Services support matrix](#).
3. To know more about the user interface of IBM Storage Fusion, see [Knowing your IBM Storage Fusion user interface](#).

Installing IBM Storage Fusion on Amazon Web Services

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

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 [IBM Storage Fusion Services support matrix](#).
- Configure image pull secrets. For more information about pull secret prerequisite, see [Creating image pull secret](#).

About this task

Support is available for only one instance of IBM Storage 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 Storage 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 Storage Fusion`.
5. Click IBM Storage Fusion.
The Version, Capability level, Source, and Provider type of IBM Storage Fusion is available.
6. Click Install.
It opens the Install Operator page for `IBM Storage 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 Storage 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.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Storage 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 Storage Fusion dialog box gets displayed.
15. In the IBM Storage 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 Storage Fusion installation](#).
 2. If you want to enable the services from the IBM Storage Fusion user interface, see the following procedures accordingly:
 - a. For Data Cataloging, see [Data Cataloging](#).
 - b. For Data Foundation, see [Data Foundation](#).
 - c. For Backup & Restore, see [Backup & Restore](#).
- If you want to install IBM Storage Fusion services using OpenShift Container Platform, then see [Deploying services from OpenShift Container Platform](#).
Important: To know more about the supported services for your platform, see [IBM Storage Fusion Services support matrix](#).
3. To know more about the user interface of IBM Storage Fusion, see [Knowing your IBM Storage Fusion user interface](#).

Installing IBM Storage Fusion on Microsoft Azure

Procedure to install IBM Storage 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 [IBM Storage Fusion Services support matrix](#).

About this task

Support is available for only one instance of IBM Storage 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 Storage 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 Storage Fusion`.
5. Click IBM Storage Fusion.
The Version, Capability level, Source, and Provider type of IBM Storage Fusion is available.
6. Click Install.
It opens the Install Operator page for `IBM Storage 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 Storage 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.

13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Storage 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 Storage Fusion dialog box gets displayed.
15. In the IBM Storage 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 Storage Fusion installation](#).
2. If you want to enable the services from the IBM Storage Fusion user interface, see the following procedures accordingly:

- a. For Data Cataloging, see [Data Cataloging](#).
- b. For Data Foundation, see [Data Foundation](#).
- c. For Backup & Restore, see [Backup & Restore](#).

If you want to install IBM Storage Fusion services using OpenShift Container Platform, then see [Deploying services from OpenShift Container Platform](#).

Important: To know more about the supported services for your platform, see [IBM Storage Fusion Services support matrix](#).

- 3. To know more about the user interface of IBM Storage Fusion, see [Knowing your IBM Storage Fusion user interface](#).

Installing IBM Storage Fusion on On-premises IBM Power Systems

Procedure to install IBM Storage 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 Storage Fusion, see [Enterprise registry for IBM Storage 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 [IBM Storage Fusion Services support matrix](#).

About this task

Support is available for only one instance of IBM Storage 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 Storage 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 Storage Fusion**.
5. Click IBM Storage Fusion.
The Version, Capability level, Source, and Provider type of IBM Storage Fusion is available.
6. Click Install.
It opens the Install Operator page for **IBM Storage 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 Storage 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.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Storage 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 Storage Fusion dialog box gets displayed.
15. In the IBM Storage 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 Storage Fusion installation](#).
 2. If you want to enable the services from the IBM Storage Fusion user interface, see the following procedures accordingly:
 - a. For Data Cataloging, see [Data Cataloging](#).
 - b. For Data Foundation, see [Data Foundation](#).
 - c. For Backup & Restore, see [Backup & Restore](#).
- If you want to install IBM Storage Fusion services using OpenShift Container Platform, then see [Deploying services from OpenShift Container Platform](#).
- Important: To know more about the supported services for your platform, see [IBM Storage Fusion Services support matrix](#).
3. To know more about the user interface of IBM Storage Fusion, see [Knowing your IBM Storage Fusion user interface](#).

Installing IBM Storage Fusion on Google Cloud

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

Before you begin

- If you plan to do an offline installation of IBM Storage Fusion, see [Enterprise registry for IBM Storage Fusion installation](#).
- For more information about deployments and their supported services, see [IBM Storage Fusion Services support matrix](#).

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 Storage 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 Storage 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 Storage Fusion**.
5. Click IBM Storage Fusion.
The Version, Capability level, Source, and Provider type of IBM Storage Fusion is available.
6. Click Install.
It opens the Install Operator page for **IBM Storage 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 Storage 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.
13. From the Applications menu in the title bar of OpenShift Container Platform, click IBM Storage 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 Storage Fusion dialog box gets displayed.
15. In the IBM Storage 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 Storage Fusion installation](#).
2. If you want to enable the services from the IBM Storage Fusion user interface, see the following procedures accordingly:
 - a. For Data Cataloging, see [Data Cataloging](#).
 - b. For Data Foundation, see [Data Foundation](#).
 - c. For Backup & Restore, see [Backup & Restore](#).If you want to install IBM Storage Fusion services using OpenShift Container Platform, then see [Deploying services from OpenShift Container Platform](#).
Important: To know more about the supported services for your platform, see [IBM Storage Fusion Services support matrix](#).
3. To know more about the user interface of IBM Storage Fusion, see [Knowing your IBM Storage Fusion user interface](#).

Installing IBM Storage Fusion from enterprise registry

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

- [Installing IBM Storage Fusion On-premises](#)
Procedure to install IBM Storage Fusion on OpenShift Container Platform.

Installing IBM Storage Fusion On-premises

Procedure to install IBM Storage 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 Storage Fusion, mirror IBM Storage 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 Storage Fusion installation](#).
- To enable IBM Fusion Data Foundation deployed on OpenShift Container Platform version 4.13, ensure a `catalogsource` named `redhat-operators` exists in `openshift-marketplace` with IBM 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:
  image: <redhat-operator catalog image available in your enterprise registry>
  sourceType: grpc
```

Example value:

```
registryhost.com:443/fusion-mirror/redhat-operator-index:v4.13
```

About this task

Support is available for only one instance of IBM Storage 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 Storage 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:2.7.0
```

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 Storage 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 Storage Fusion user interface. In the OpenShift Container Platform web console, click the Applications icon in the title bar and select IBM Storage 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 Storage 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 Storage Fusion, see [Managing services](#).

Validating IBM Storage Fusion installation

Steps to confirm the success of the installation.

- [Health status of individual services](#)
- [Verify the IBM Storage Fusion installation](#)
- [Verify the Storage or Global data platform installation](#)
- [Verify the Backup & Restore installation](#)
- [Verify the IBM Fusion Data Foundation](#)
- [Verify the Data Cataloging installation](#)
- [Verify the IBM Storage 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 Storage 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 Storage 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-6d8754cf4-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-5bbcdb786b-hhzsj                   1/1     Running   0          134m
isf-serviceability-operator-controller-manager-54c555676c-714pq 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-1ppmg                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 Storage Fusion instances as follows:

1. Click Operators->Install operators. Select the namespace in which IBM Storage Fusion was installed from the Project tab.
2. Select the IBM Storage Fusion.
3. Go to All instance tab. It lists the IBM Storage 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 Storage 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 Storage Fusion user interface in the OpenShift Container Platform web console, click the Applications icon in the title bar and select IBM Storage Fusion.
Note: You can use IBM Storage Fusion user interface only when the installation of IBM Storage Fusion is completed successfully.

Verify the Storage or Global data platform installation

Note:

- Ensure that the Global data platform service is enabled before validation.
- IBM Storage Fusion 2.8.0 or higher supports IBM Storage Scale 5.2.0.

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 Storage 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 Storage Fusion UI.

2. Verify the IBM Storage Scale deployments and pods from the IBM Storage 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:

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-5d5fbdfbb-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 Storage 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 Storage Fusion service is enabled before validation.

Verify that the Backup & Restore IBM Storage 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 Storage Fusion Backup and Restore Server
- IBM Storage Fusion Backup and Restore Agent
- OADP Operator

4. Alternatively, you can verify the status of operators by running the following oc command:

Note: Note that 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.v2.6.0-2	AMQ Streams	2.6.0-2	
amqstreams.v2.6.0-1	Succeeded		
guardian-dm-operator.v2.8.0	IBM Storage Fusion Backup and Restore Data Mover	2.8.0	
Succeeded			
guardian-dp-operator.v2.8.0	IBM Storage Fusion Backup and Restore Data Protection	2.8.0	
Succeeded			
guardian-mongo-operator.v2.8.0	IBM Storage Fusion Backup and Restore Mongo	2.8.0	
Succeeded			
ibm-dataprotectionagent.v2.8.0	IBM Storage Fusion Backup and Restore Agent	2.8.0	
Succeeded			
ibm-dataprotectionserver.v2.8.0	IBM Storage 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 Storage Fusion Backup and Restore Redis	2.8.0	
Succeeded			

Make sure 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 Storage 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
b0f64f9161e0882f278dde2ea1ea9677f4a230a29180fcf21fc665761hvxxz	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
bc6176f08ef686cde24395724b77ea07a586a0bd1fa27ebfd5d704d0dxv9pl	0/1	Completed	0
e349f7c16f02ad6c0c31e41ba2fb1750d5154b58537224d239fe47508872cj5	0/1	Completed	0
f3dd0cbe8cb98614cf163fc5372733148236ea2bdeb5efc6a5d5afe4c085qlk	0/1	Completed	0
ff53c6d827e0fd610a4392c4f941beb0c785572d2fcalbda57e208650bwv2m	0/1	Completed	0
ffacc5cda1e0aa4f3b3c0021f3c87931aa7422f8415303d95457febcb8nmzk7	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 Storage 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 Storage 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.
 - Make sure 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 IBM Fusion Data Foundation

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

1. Go to Installed Operators to check whether the IBM Fusion Data Foundation operator is listed and status is Succeeded.
2. Verify pods from the Red Hat®OpenShift Container Platform web console:

- a. Go to Workloads > Pods.
- b. Select the `openshift-storage` namespace, where the IBM Fusion Data Foundation is installed.
- c. Check whether the following pods are running successfully:
 - `csi-addons-controller-manager`
 - `noobaa-operator`
 - `ocs-metrics-exporter`
 - `ocs-operator`
 - `odf-console`
 - `odf-operator-controller-manager`
 - `rook-ceph-operator`

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
<code>csi-addons-controller-manager-bdb965b47-jvfhs</code>	2/2	Running	0	119s
<code>noobaa-operator-746b6bc54-6541h</code>	1/1	Running	0	3m3s
<code>ocs-metrics-exporter-69cf56fd9-jk49s</code>	1/1	Running	0	2m45s
<code>ocs-operator-6879c74556-pppj9</code>	1/1	Running	0	2m46s
<code>odf-console-849f64fdd7-rqd97</code>	1/1	Running	0	3m3s
<code>odf-operator-controller-manager-5bd4c85d6b-q8g2f</code>	2/2	Running	0	3m3s
<code>rook-ceph-operator-76699f976c-2zpmz</code>	1/1	Running	0	2m46s

4. Ensure that all the pods are up and running.

5. Verify the IBM Fusion Data Foundation status as follows:

- a. Click Operators > Install operators. Select the namespace in which IBM Storage Fusion was installed from the Project tab.
- b. Select the IBM Storage 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 Storage Fusion was installed from the Project tab.
- b. Select the IBM Storage 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:

```
[root@fu40 ~]# oc get pod -n openshift-local-storage
NAME                               READY   STATUS    RESTARTS   AGE
local-storage-operator-75c55cf57d-pfjcp   1/1     Running   0          109s
```

Verify the Data Cataloging installation

Verify the 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 Data Cataloging status.

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

Expected output	READY	SUMMARY	ERROR	AGE	HEALTHSTATUS
NAME					
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
activevelogs-c-isd-db2u-0	Bound	pvc-7e98eeead-9a38-48b6-a880-16d730120796	150Gi	RWO		ocs-storageclass
activevelogs-c-isd-db2u-1	Bound	pvc-7af71786-1529-4b3c-bbd1-b5d1ee733d33	150Gi	RWO		ocs-storageclass
c-isd-meta	Bound	pvc-1ba20699-ddb8-4600-bf9f-4d637b14da49	50Gi	RWX		ocs-storageclass
data-c-isd-db2u-0	Bound	pvc-488540db-c951-4f48-b65c-814bf634ca33	300Gi	RWO		ocs-storageclass
data-c-isd-db2u-1	Bound	pvc-56a3105a-347e-4b55-b11a-d0a1d351d622	300Gi	RWO		ocs-storageclass
data-isd-sasl-kafka-0	Bound	pvc-3924c9cc-173d-4244-90fa-a234325dc04b	100Gi	RWO		ocs-storageclass
data-isd-sasl-zookeeper-0	Bound	pvc-3c66397b-ac03-4ed8-841a-e33d83355a23	64Gi	RWO		ocs-storageclass
data-isd-ssl-kafka-0	Bound	pvc-ce0ebef-d2b6-4e40-8c31-e31d45a208e5	100Gi	RWO		ocs-storageclass
data-isd-ssl-zookeeper-0	Bound	pvc-cd3fdd20-343b-49ca-ad1a-690f8eec3e60	64Gi	RWO		ocs-storageclass
isd-backup	Bound	pvc-2e077c8e-e992-4e1a-8b98-d24bfbdb5343	250Gi	RWX		ocs-storageclass
isd-data	Bound	pvc-b98e124f-1f98-43e5-9bf6-c8758c681192	250Gi	RWX		ocs-storageclass
temptsc-isd-db2u-0	Bound	pvc-7fdc9ac5-9158-48ac-bad3-3c24590f1f2b	50Gi	RWO		ocs-storageclass
temptsc-isd-db2u-1	Bound	pvc-a2adb668-a362-4f61-8ed3-ae103f9b2cd3	50Gi	RWO		ocs-storageclass

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	c-isd-sshkeys-db2uhausr	OK
secret	c-isd-sshkeys-db2uadm	c-isd-sshkeys-db2instusr	OK
secret	c-isd-ldappassword	c-isd-ldapblueadminpassword	OK
secret	c-isd-instancepassword	c-isd-db2u-lic	OK
secret	c-isd-certs-wv-rest	c-isd-certs-db2u-api	OK
configmap	c-isd-db2uconfig	c-isd-db2regconfig	OK
configmap	c-isd-db2dbmconfig	c-isd-db2dbcconfig	OK
configmap	c-isd-ldap	c-isd-tools	OK
Deployment	c-isd-tools	c-isd-db2u-api	OK
service	c-isd-db2u-engn-svc	c-isd-db2u-head-engn-svc	OK
service	c-isd-db2u-rest-svc	c-isd-instdb	OK
Job	c-isd-etcd	c-isd-etcd	OK
service	c-isd-rest-ext	c-isd-db2u	OK
Deployment	c-isd-rest	c-isd-graph-ext	OK
service	c-isd-db2u-graph-svc	c-isd-graph	OK
networkpolicy	c-isd-db2u-internal	c-isd-db2u-rest-morph	OK
Deployment	c-isd-db2u	c-isd-db2u	OK
service	c-isd-db2u-ext	c-isd-db2u	OK
StatefulSet	c-isd-db2u	c-isd-db2u	OK
service	c-isd-db2u	c-isd-db2u	OK
networkpolicy	c-isd-db2u	c-isd-db2u	OK
networkpolicy	c-isd-db2u	c-isd-db2u	OK
Job	c-isd-db2u	c-isd-db2u	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-5fc8bd95f-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-5f9b5d4676-178cc	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-fq55m	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-lvw15	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-b5bdv	1/1	Running	0	9d
isd-consumer-file-scan-5c7565bf7b-b5l1v2	1/1	Running	0	9d
isd-consumer-file-scan-5c7565bf7b-d9gbm	1/1	Running	0	9d
isd-consumer-file-scan-5c7565bf7b-f5rxr	1/1	Running	0	9d
isd-consumer-file-scan-5c7565bf7b-m8nw6	1/1	Running	0	9d

isd-consumer-file-scan-5c7565bf7b-ntfb	1/1	Running	0	9d
isd-consumer-file-scan-5c7565bf7b-ql55x	1/1	Running	0	9d
isd-consumer-file-scan-5c7565bf7b-qvgm	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-q8qgf	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-7bdc7957c6-981m9	1/1	Running	0	9d
isd-consumer-scale-scan-7bdc7957c6-d4z1j	1/1	Running	0	9d
isd-consumer-scale-scan-7bdc7957c6-fc82j	1/1	Running	0	9d
isd-consumer-scale-scan-7bdc7957c6-mnns2	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-5d7bfb887-r487x	1/1	Running	1 (9d ago)	9d
isd-db-schema-xm5cn	0/1	Completed	0	9d
isd-db2whrest-6c6bbfc6c-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-44fwf	1/1	Running	0	9d
isd-reports-c6kmq	0/1	Completed	0	9d
isd-scaleafmdata mover-85cc44b566-644vx	1/1	Running	1 (9d ago)	9d
isd-scaleilmdata mover-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>	reenrypt	None

Verify the IBM Storage Fusion CR

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

1. Important: The IBM Storage Fusion namespace **ibm-spectrum-fusion-ns** is used in the following steps for your reference. If IBM Storage Fusion is installed in a different namespace, replace the namespace with your namespace.

Run the following command and export IBM Storage Fusion namespace as environmental variable.

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

2. Run the command to validate the IBM Storage 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. Navigate to OpenShift Container Platform console, go to Operators → Installed operators.

4. Select the IBM Storage Fusion namespace from the Project drop down and select the Storage Fusion tab.

It shows the CR kind and name of the CR.

5. This CR show status and version of all IBM Storage Fusion services as well as the IBM Storage 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 Storage Fusion

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

Before you begin

- Uninstall all IBM Storage Fusion services before you uninstall IBM Storage Fusion. For the procedure, see [Uninstalling services](#).
- Ensure that you delete all the connection CRDs. For the procedure to delete, see [Uninstalling services](#).

Procedure

1. Run the following commands to delete the CRDs and related objects:

```

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

oc delete sppmanagers sppmanager
oc delete updatemanager version

oc delete crd fusionservicedefinitions.service.isf.ibm.com
oc delete crd fusionserviceinstances.service.isf.ibm.com

oc delete crd odfmanagers.odf.isf.ibm.com
oc delete crd odfclusters.odf.isf.ibm.com
oc delete crd applications.application.isf.ibm.com
oc delete crd clusters.application.isf.ibm.com
oc delete crd connections.application.isf.ibm.com
oc delete crd backuppolicies.data-protection.isf.ibm.com
oc delete crd backups.data-protection.isf.ibm.com
oc delete crd backupstoragelocations.data-protection.isf.ibm.com
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 controlplanebackups.bkprstr.isf.ibm.com
oc delete crd csiscaleoperators.cs.ibm.com
oc delete crd daemons.scale.spectrum.ibm.com
oc delete crd deletebackuprequests.data-protection.isf.ibm.com
oc delete crd encryptionclients.cns.isf.ibm.com
oc delete crd encryptionconfigs.scale.spectrum.ibm.com
oc delete crd encryptionservers.cns.isf.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 hooks.data-protection.isf.ibm.com
oc delete crd ibmsppcs.sppc.ibm.com
oc delete crd ibmsppps.ocp.spp.ibm.com
oc delete crd isfcdeps.mgmtsft.isf.ibm.com
oc delete crd isfemdeps.mgmtsft.isf.ibm.com
oc delete crd isflcdeps.mgmtsft.isf.ibm.com
oc delete crd isfproxydeps.mgmtsft.isf.ibm.com
oc delete crd isfuideps.mgmtsft.isf.ibm.com
oc delete crd isfconsoleplugins.mgmtsft.isf.ibm.com
oc delete crd localdisks.scale.spectrum.ibm.com
oc delete crd pmcollectors.scale.spectrum.ibm.com
oc delete crd policyassignments.data-protection.isf.ibm.com
oc delete crd recoverygroups.scale.spectrum.ibm.com
oc delete crd remoteclusters.scale.spectrum.ibm.com
oc delete crd restores.data-protection.isf.ibm.com
oc delete crd scaleclusters.cns.isf.ibm.com
oc delete crd scalemanagers.cns.isf.ibm.com
oc delete crd spectrumfusions.prereq.isf.ibm.com
oc delete crd sppmanagers.spp.isf.ibm.com
oc delete crd updatemanagers.update.isf.ibm.com

```

```

oc delete crd computeprovisionworkers.install.isf.ibm.com
oc delete crd storagesetUps.install.isf.ibm.com
oc delete crd nodeconfigs.monitor.isf.ibm.com
oc delete crd fusionServiceDefinitions.service.isf.ibm.com
oc delete crd fusionServiceInstances.service.isf.ibm.com
oc delete crd recipes.spp-data-protection.isf.ibm.com
oc delete crd sncFilesystems.cns.isf.ibm.com
oc delete crd sncNodes.cns.isf.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 stripePEFSJobs.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

```

2. Log in to Red Hat OpenShift Container Platform console, go to Operators > Installed Operators.
3. Select IBM Storage Fusion > Uninstall Operator.
4. Export IBM Storage Fusion namespace as an environmental variable.

Note: If the IBM Storage Fusion is installed in a different namespace, replace the existing namespace:

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

5. Run the following command to delete the IBM Storage Fusion namespace:

```
oc delete ns ${FUSION_NS}
```

6. Run the following commands to clean the manifests:

```

oc delete clusterrole isf-application-operator-role-connection-kube-config
oc delete clusterrolebinding isf-application-operator-rolebinding-connection-kube-config

```

Deploying IBM Storage Fusion services

In the service-based IBM Storage Fusion, choose the features that you want to deploy. You do not have to install all the services at installation time instead enable a feature that is available as a service on-demand basis.

As an administrator, you can install or enable the required services from the user interface, but make sure you meet the resource requirements of individual services.

Global Data Platform service

The Global Data Platform storage type provides the following features:

- File storage
- High availability via 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.

Data Foundation

IBM Fusion Data Foundation as storage in IBM Storage Fusion. You can enable and manage the IBM Fusion Data Foundation services, deploy, and scale up or out the storage cluster. The IBM 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 IBM Fusion Data Foundation service, see [Introduction to Fusion Data Foundation](#).

Backup & Restore

The Backup & Restore service provides the following features:

- Policy-driven backup of applications that run on Red Hat® OpenShift®
- 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. To know more about the service and its architecture, see [Backup and Restore](#).

Data Cataloging

The Data Cataloging provides data insight for exabyte-scale heterogeneous file, object, backup, and archive storage on premises and in the cloud. The software easily connects to these data sources to rapidly ingest, consolidate, and index metadata for billions of files and objects. For more information about the Data Cataloging service and its architecture, see [Data Cataloging](#).

You can also manage the upgrade of these IBM Storage Fusion services from the user interface. For steps to upgrade in IBM Storage Fusion, see [Upgrading IBM Storage Fusion services](#).

- [IBM Storage Fusion Services support matrix](#)

Support matrix of IBM Storage Fusion services on the different deployment platforms of IBM Storage FusionSupport matrix of IBM Storage Fusion services on the Bare Metal deployment platform of IBM Storage Fusion HCI System.

- [Installing services](#)
Install services from the user interface of IBM Storage Fusion.
- [Uninstalling services](#)
You can uninstall IBM Storage Fusion services by using the command line.

IBM Storage Fusion Services support matrix

Support matrix of IBM Storage Fusion services on the different deployment platforms of IBM Storage Fusion Support matrix of IBM Storage Fusion services on the Bare Metal deployment platform of IBM Storage Fusion HCI System.

IBM Storage Fusion

Table having a matrix of what IBM Storage Fusion service is supported on which deployment platforms of IBM Storage Fusion

Amazon Web Services

IBM Storage Fusion Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
Self-managed OpenShift® Container Platform on Amazon Web Services	Yes	Yes	No	Yes	Yes
Amazon Web Services ROSA	No	No	Yes	Yes	Yes

Microsoft Azure

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
Self-managed OpenShift Container Platform on Microsoft Azure	Yes	Yes	No	Yes	Yes
Microsoft Azure ARO	Yes	Yes	No	Yes	Yes

IBM Cloud®

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
ROKS IBM Classic	No	Yes	No	Yes	Yes
ROKS VPC	No	Yes	No	Yes	Yes

IBM Power Systems

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
IBM Power Systems	Yes	Yes	Yes	Yes	Yes

Bare Metal

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
Bare Metal	Yes	Yes	Yes	Yes	Yes

Linux on IBM zSystems

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
Linux on IBM zSystems	Yes	Yes	Yes	Yes	No

zCX

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
zCX	Yes	Yes	Yes	Yes	Yes

On-premises VMware

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
On-premises VMware	Yes	Yes	Yes	Yes	Yes

Google Cloud

Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
Google Cloud	Yes	Yes	No	Yes	No

Installing services

Install services from the user interface of IBM Storage Fusion.

Before you begin

- In the Get started section of the Services page, you must first install and configure Data Foundation service or Global Data Platform service as your storage. You can install Additional services only after you configure storage.

About this task

If you want to install IBM Storage Fusion services using OpenShift® Container Platform console, then see [Deploying services from OpenShift Container Platform](#).

- [Deploying services from OpenShift Container Platform](#)
You can deploy services also from the OpenShift Container Platform console.
- [Global Data Platform](#)
From the Services page, enable the IBM Storage Fusion Global data platform service.
- [Data Foundation](#)
The service provides a foundational data layer for applications to function and interact with data in a simplified, consistent, and scalable manner.
- [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 Storage Fusion Backup & Restore service installation include backup hub and backup spoke.
- [Data Cataloging](#)
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.

Deploying services from OpenShift Container Platform

You can deploy services also from the OpenShift® Container Platform console.

Procedure

1. In the OpenShift Container Platform web console, go to Installed Operators > IBM Storage Fusion > Fusion Service instance tab.
2. Create the `FusionServiceInstance` object for the respective service with the following specifications:

Data Foundation

```
apiVersion: service.isf.ibm.com/v1
kind: FusionServiceInstance
metadata:
  name: odfmanager
  namespace: ibm-spectrum-fusion-ns
spec:
  creator: User
  doInstall: false
  parameters:
    - name: namespace
      provided: true
      value: ''
    - name: creator
      provided: true
      value: Fusion
    - name: backingStorageType
      provided: true
      value: Local
    - name: autoUpgrade
      provided: true
      value: 'true'
  serviceDefinition: data-foundation-service
  triggerUpdate: false
```

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 [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.

Data Cataloging

```
apiVersion: service.isf.ibm.com/v1
kind: FusionServiceInstance
metadata:
  name: data-cataloging-service-instance
  namespace: ibm-spectrum-fusion-ns
spec:
  creator: User
  doInstall: true
  parameters:
    - name: namespace
      provided: false
      value: ibm-data-cataloging
    - name: rwx_storage_class
      provided: true
      value: <storage-class-name>
    - name: doInstall
      provided: false
      value: 'true'
    - name: license
      provided: false
      value: '{"accept": true}'
  serviceDefinition: data-cataloging-service-definition
  triggerUpdate: false
```

Backup & Restore

Backup & Restore server:

```
apiVersion: service.isf.ibm.com/v1
```

```

kind: FusionServiceInstance
metadata:
  name: ibm-backup-restore-service-instance
  namespace: ibm-spectrum-fusion-ns
spec:
  creator: User
  doInstall: true
  parameters:
    - name: namespace
      provided: false
      value: ibm-backup-restore
    - name: storageClass
      provided: true
      value:<storage-class-name>
    - name: doInstall
      provided: false
      value: 'true'
  serviceDefinition: ibm-backup-restore-service
  triggerUpdate: false

```

Backup & Restore Agent:

```

apiVersion: service.isf.ibm.com/v1
kind: FusionServiceInstance
metadata:
  name: dpagent
  namespace: ibm-spectrum-fusion-ns
spec:
  creator: Fusion
  doInstall: true
  parameters:
    - name: namespace
      provided: true
      value: ibm-backup-restore
    - name: storageClass
      provided: true
      value: <storage-class-name>
  serviceDefinition: ibm-backup-restore-agent-service
  triggerUpdate: false

```

Global Data Platform

From the Services page, enable the IBM Storage Fusion Global data platform service.

About this task

Important:

Warning: 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.
- For more information about deployments and their supported IBM Storage Fusion services in IBM Storage Fusion, see [IBM Storage Fusion Services support matrix](#).
- 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](#).

Procedure

1. Go to the Services page in the IBM Storage 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
Healthy	The Running and Ready states for the following items indicate that the service is healthy: <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • 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 Storage Fusion, see [Troubleshooting installation and upgrade issues in IBM Storage Fusion services](#).

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. Go to the 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](#).
 3. You can now install other available IBM Storage Fusion services.
- [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

- 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` 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` 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>
```

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-xxxx.yaml": admission webhook "mscalecluster.fusion.s" did not allow the operation: invalid character '-' at index 10: yaml: line 1:10: did not expect '-' in scalar

3. Apply the `Scalecluster` CR.
4. Monitor the filesystem and cluster status on the IBM Storage Fusion user interface.

Data Foundation

The service provides a foundational data layer for applications to function and interact with data in a simplified, consistent, and scalable manner.

- **Data Foundation**

Install Fusion Data Foundation in local, dynamic, external, or consumer modes.

Data Foundation

Install Fusion Data Foundation in local, dynamic, external, or consumer modes.

Before you begin

- If you already have an installation of IBM Fusion Data Foundation cluster or Red Hat® OpenShift® Data Foundation cluster, then the IBM Storage Fusion UI automatically discovers it. However, in the Data Foundation page, you can only view the Usable capacity, Health, and information about the storage nodes.
- If you have a Rook-Ceph operator, then contact [IBM Support](#) for the expected behavior.
- You can use a maximum of nine storage devices per node. The high number of storage devices leads to a higher recovery time during the loss of a node. This recommendation ensures that nodes stay below the cloud provider dynamic storage device attachment limits, and limits the recovery time after node failure with local storage devices.

It is recommended to add nodes in the multiple of three, each of them in different failure domains.

For deployments having three failure domains, you can scale up the storage by adding disks in the multiple of three, with the same number of disks coming from nodes in each of the failure domains.

- Run following steps to ensure that there is no previously installed IBM Fusion Data Foundation, Red Hat OpenShift Data Foundation, OpenShift Container Storage, or Rook-Ceph in this OpenShift cluster before you enable the IBM Fusion Data Foundation service:

1. Verify that the namespace `openshift-storage` does not exist.
2. Run the following command to confirm that no resources exist:

```
oc get storagecluster -A; oc get cephcluster -A
```

3. Run the following command to confirm that none of the nodes have a IBM Fusion Data Foundation storage label or Red Hat OpenShift Data Foundation storage label:

```
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 of IBM Fusion Data Foundation](#).
- For IBM Fusion Data Foundation resource requirements, see [ODF sizer tool](#). For common IBM Fusion Data Foundation components per instance requirements of CPU and Memory, see [Resource requirements](#).
- If you want to deploy IBM 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.

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 Storage Fusion, you cannot add nodes or capacity from IBM Storage Fusion. See [Red Hat OpenShift Data Foundation Scaling storage document](#).
- For consumer mode cluster with Fusion Data Foundation, you do not install IBM Storage Fusion spoke cluster or Fusion Data Foundation from the user interface. These installations are automatically run from the IBM Storage Fusion HCI System hub cluster.
- Data Foundation service supports four storage types: Local, Dynamic, External, and consumer. Only one device type can be used for Data Foundation service in an OpenShift Container Platform. The following table shows supported platforms and which device types are supported in each platform.

Platform	Support
VMware	Local, dynamic, and external device
Bare Metal	Local, external device, and consumer mode
Linux on IBM zSystems	Local, dynamic, and external device
IBM Power Systems	Local and external device
ROKS on IBM Cloud (ROKS on VPC and Classic Bare Metal)	Cloud-managed IBM Fusion Data Foundation service. For the procedure to install, see Understanding IBM Fusion Data Foundation .
Self managed OpenShift Container Platform on Microsoft Azure	Dynamite storage class
Self managed OpenShift Container Platform on Amazon Web Services	Dynamic storage class
Self managed OpenShift Container Platform on Google Cloud	Dynamic storage class

- For IBM Storage Fusion services and platform support matrix, see [IBM Storage Fusion Services support matrix](#).

Procedure

1. Go to Services page in IBM Storage Fusion user interface.
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 consumer, 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 Storage Fusion auto upgrades IBM Fusion Data Foundation version.

This upgrades IBM Fusion Data Foundation service within the subscription channel like 4.10.x to 4.10.y, and across the subscription channel like 4.10.x to 4.11.y. However, it does not auto convert Red Hat OpenShift Data Foundation to IBM Storage Fusion Data Foundation.

For the procedure to upgrade Red Hat OpenShift Data Foundation 4.12 to IBM Fusion Data Foundation 4.12 or higher, see [Upgrading Red Hat OpenShift Data Foundation 4.12 to IBM Storage Fusion Data Foundation 4.12 or higher](#).

The IBM 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 IBM Fusion Data Foundation operator is listed and Status shows as Succeeded.
- Additionally, if the installed platform is Bare Metal, Linux on IBM zSystems, or VMware using local device, check whether the `Local Storage operators` is installed and its status shows as Succeeded.

7. Verify the IBM Fusion Data Foundation service installation.

- a. To verify the installation of IBM Fusion Data Foundation on IBM Storage Fusion, perform the checks specified in [Verify the IBM Fusion Data Foundation](#).
- b. Additionally, run the command to verify that IBM Fusion Data Foundation service is installed successfully:

```
oc describe odfmanager/odfmanager -n <Fusionns>
```

Update `<Fusions>` with your namespace value.

Table 1. Install states IBM Fusion Data Foundation service

Install State	Description
<code>Installing</code>	IBM Fusion Data Foundation installation is ongoing, and there are no errors yet.
<code>Failing</code>	There is an installation error but IBM Storage Fusion retries to install IBM Fusion Data Foundation. If the problem is solved, this status changes to <code>Installing</code> or <code>Completed</code> .
<code>Completed</code>	IBM Fusion Data Foundation installation completed successfully.

Table 2. Upgrade states IBM Fusion Data Foundation service

Upgrade State	Description
<code>Not started</code>	IBM Fusion Data Foundation upgrade has not started yet.
<code>Upgrading</code>	IBM Fusion Data Foundation upgrade is ongoing, and there are no errors yet.
<code>Failing</code>	There is an installation error but IBM Storage Fusion retries to upgrade IBM Fusion Data Foundation. If the problem is solved, this status changes to <code>Upgrading</code> or <code>Completed</code> .
<code>Completed</code>	IBM Fusion Data Foundation upgrade completed successfully.

Table 3. Health states IBM Fusion Data Foundation service

State	Description
<code>Installing</code>	IBM Fusion Data Foundation installation is ongoing. For more information about installation status details, see Table 1 .
<code>Upgrading</code>	IBM Fusion Data Foundation upgrade is ongoing. For more information about upgrading status details, see Table 2 .
<code>Healthy</code>	IBM Fusion Data Foundation installation completed successfully and the service is in normal state.
<code>Degraded</code>	IBM Fusion Data Foundation installation completed successfully but the service is not in normal state.

What to do next

To install and work with other dependent services, configure storage in your environment. Provision the amount of usable capacity that your environment needs and select the nodes to run the Data Foundation workloads. For more information about the configuration, see [Configuring Data Foundation storage](#).

Optionally, you 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 IBM Fusion Data Foundation](#).

Note: You can view the Usable capacity and Health for an externally discovered IBM Fusion Data Foundation but cannot configure storage.

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 Storage Fusion Backup & Restore service installation include backup hub and backup spoke.

About this task

To know more about the hub and spoke architecture of Backup & Restore service, see [Hub and spoke model overview](#).

- **[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.

- [Disabling the connection](#)

Disable the connection between Hub and Spoke.

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

- Spoke

- Must be able to make a TCP connection to the Hub cluster API address
 - Must be able to make a TCP connection address `<kafka-route>:443`, where kafka-route can be found by running the following command on the hub:

```
oc get route kafka-bridge-rbac-proxy -n ibm-backup-restore -o template --template  
'{{.spec.host}}'
```

- 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
```

Procedure

1. Go to Services page in IBM Storage 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.

The internal data catalog requires a minimum of 200 GB for ReadWriteOnce storage so select a storage class that supports this criteria.

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:

- Environment
- 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 Storage Fusion, see [Troubleshooting installation and upgrade issues in IBM Storage Fusion services](#).

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 Storage 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 YAML is 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
 - Spoke
 - Must be able to make a TCP connection to the Hub cluster API address
 - Must be able to make a TCP connection address <kafka-route>:443, where kafka-route can be found by running the following command on the hub:

```
oc get route kafka-bridge-rbac-proxy -n ibm-backup-restore -o template --template
'{.spec.host}'
```

- 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
```

Procedure

1. Go to Services page in IBM Storage 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.
The internal data catalog requires a minimum of 200 GB for ReadWriteOnce storage so select a storage class that supports this criteria.
 5. 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 Storage Fusion user interface.
 6. 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:
- Environment
 - 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 Storage Fusion, see [Troubleshooting installation and upgrade issues in IBM Storage Fusion services](#).

What to do next

- You can set the configuration parameters in ConfigMap guardian-configmap to change defaults for IBM Storage 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 Storage 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
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.
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.

4. To validate the connection, do the following steps:
 - a. Click Backup & Restore > Environment.
In the Environment page, you can see the hub cluster. In the Cluster table, you can see a list of spokes connected.
 - b. In the Environment 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.

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.

Method 1: Using command line or command prompt

Procedure

- 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
```

- 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

- Log in to OpenShift® Container Platform console.
- Go to Administration > CustomResourceDefinitions > connections.application.isf.ibm.com > Instances.
- Get the connection instance whose cluster name in the annotation is the same as the hub cluster name.
- Delete this connection instance.

Data Cataloging

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 Data Cataloging service.
- The following details are a base line for finding the resources that are needed for IBM Storage Fusion 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 Storage Fusion Data Cataloging service must have dedicated compute resources. Make sure 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	500 GB	500 M

- The standard deployment for Data Cataloging service project requests and limits:

Table 2. OpenShift Container

Platform requests and limits

Custom resources	Limits
CPU requests	13400 m
CPU limits	76500 m
Memory requests	27278 Mi
Memory limits	153628 Mi

- Important: For the 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
- Go through troubleshooting information related to the installation of Data Cataloging. See [Data Cataloging service issues](#).

About this task

Important: If you have OpenShift® Container Platform version 4.15, then you cannot install the Data Cataloging service.

Procedure

- Go to Services page in IBM Storage Fusion user interface.
- In the Available section, click Data cataloging tile.
- In the Data cataloging window, go through the details of the service and click Install.
- 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.

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 Storage Fusion, see [Troubleshooting installation and upgrade issues in IBM Storage Fusion services](#).

6. Validate the installation.

- IBM Storage Fusion user interface:

After you enable the 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 3. Health states 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

Uninstalling services

You can uninstall IBM Storage Fusion services by using the command line.

As a prerequisite to uninstallation of services, disable connections that are setup. For the actual procedure to disable, see [Disabling connections](#).

Important: If you want to do a complete uninstallation, then uninstall all your installed services in the sequence mentioned in this topic and then uninstall IBM Storage Fusion.

- [Disabling connections](#)

Before you uninstall, disable the connection by using command-line or command prompt.

- [Uninstalling Data Cataloging](#)

Steps to uninstall Data Cataloging by using command line interface.

- [Uninstalling Backup & Restore](#)

Steps to uninstall IBM Storage Fusion Backup & Restore by using command-line or command prompt interface.

- [Uninstalling Data Foundation](#)

Use the following steps to uninstall Data Foundation by using command line interface.

- [Uninstalling Global Data Platform](#)

Steps to uninstall Global Data Platform by using command line interface.

Disabling connections

Before you uninstall, disable the connection by using command-line or command prompt.

Procedure

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

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

2. Run the following `oc` command to list all the connection CR:

```
oc get connections -n "$FUSION_NS"
```

3. Run the following `oc` command to delete all the connection CRs in this cluster:

```
oc delete connections <Connection_Name> -n "$FUSION_NS"
```

Uninstalling Data Cataloging

Steps to uninstall Data Cataloging by using command line interface.

Procedure

1. Run the following command and export IBM Storage Fusion namespace as an environmental variable.

```
export FUSION_NS=<Storage Fusion namespace>"
```

Important: Replace `<Storage 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 Data Cataloging workload, networking and storage resources.

```
oc project default  
oc delete project ibm-data-cataloging
```

5. Run the following command to delete the IBM Storage Fusion 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 Data Cataloging **ConsoleLink** object.

```
oc delete consolelink/data-cataloging
```

8. Run the following command to delete Data Cataloging **CustomResourceDefinitions** from the cluster.

```
oc delete customresourcedefinition/spectrumdiscover.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
```

Uninstalling Backup & Restore

Steps to uninstall IBM Storage Fusion Backup & Restore by using command-line or command prompt interface.

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](#).

Uninstalling Data Foundation

Use the following steps to uninstall Data Foundation by using command line interface.

Procedure

1. Run the following command and export IBM Storage Fusion namespace as environmental variable.

```
export FUSION_NS=<Storage Fusion namespace>"
```

Important: Replace `<Storage Fusion namespace>` with your namespace.

2. Run the following `oc` command and scale the IBM Storage 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 IBM Storage Fusion Data Foundation CR.

```
oc delete odfmanager odfmanager  
oc delete odfcluster odfcluster -n "$FUSION_NS"
```

4. Uninstall the Data Foundation. For uninstallation steps, see <https://access.redhat.com/articles/6525111>.

5. Run the following `oc` command and scale the IBM Storage 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 IBM Storage Fusion Data Foundation Service Instance CR.

```
oc delete fusionserviceinstance odfmanager -n "$FUSION_NS"
```

7. Run the following `oc` command to delete the IBM Storage Fusion Data Foundation Catalog Source.

```
oc delete catalogsource isf-data-foundation-catalog -n openshift-marketplace
```

Uninstalling Global Data Platform

Steps to uninstall Global Data Platform by using command line interface.

Before you begin

- Run the following commands to define namespace environmental variables:

```
export FUSION_NS=<Storage Fusion namespace>
export SCALE_NS=ibm-spectrum-scale
```

Important: Replace <Storage

Fusion namespace> with your namespace.

- Make sure 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:
 - Stop all the applications that are accessing storage through the IBM Storage Scale Container Storage Interface Driver.
 - Delete all the Persistent Volume Claims (PVCs) and Persistent Volumes (PVs) provisioned by IBM Storage Scale Container Storage Interface Driver.

Procedure

- Do the following steps to delete the IBM Storage Fusion storage CRs.

- 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 {}
```

- 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 {}
```

- Run the following commands to verify all `EncryptionServer` and `EncryptionClient` CRs are deleted from IBM Storage Fusion namespace:

```
oc -n ${FUSION_NS} get EncryptionServer
oc -n ${FUSION_NS} get EncryptionClient
```

- Run the following command to verify all the `ScaleCluster` CRs are deleted from IBM Storage Fusion namespace.

```
oc -n ${FUSION_NS} get ScaleCluster
```

- 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 {}
```

- 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 {}
```

- Do the following steps to delete IBM Storage Scale container native resources.

- 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
```

- Run the following command to delete the IBM Storage Scale container native cluster:

```
oc delete cluster.scale.spectrum.ibm.com ibm-spectrum-scale
```

- 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
```

- 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 Storage Fusion Global Data Platform service:

```
oc -n ${FUSION_NS} patch spectrumfusion/spectrumfusion --type=merge --patch='{"spec":{"GlobalDataPlatform":  
{"Enable":false}}}'
```

b. Run the following command to delete ScaleManager CR:

```
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:

```
oc delete -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/generated/scale/install.yaml --ignore-not-found=true
```

Upgrading IBM Storage Fusion

Procedure to upgrade IBM Storage Fusion to 2.8.

Before you begin

- For 2.8, you must be on IBM Storage Fusion version 2.7.2.
- If you installed IBM Storage Fusion version 2.7.2 by using online or your enterprise registry installation mode, then ensure that you do not change the mode during the upgrade to 2.8 version. To change the installation mode, reinstall IBM Storage Fusion.
- Ensure all compute nodes are in ready state on OpenShift® user interface.
- Download the logs that you collected by using IBM Storage 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 Storage Fusion by using your enterprise registry, then do these additional mirroring prerequisites steps. See [Prerequisites for enterprise registry upgrade](#).

Procedure

1. Log in to the OpenShift Container Platform management console as the cluster administrator.

2. Upgrade IBM Storage 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 Storage 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 Storage Fusion is Automatic. However, you can change it to Manual.

f. After the upgrade is successful, refresh your browser and clear your cache.

g. Verify whether the IBM Storage Fusion is in succeeded state and the version is 2.8.0. Also, in the Subscription tab, ensure that the upgrade status displays Up to date.

3. Upgrade IBM Storage Fusion services.

For the steps to upgrade services, see [Upgrading IBM Storage 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 Storage Fusion. Do not upgrade or rollback your Red Hat OpenShift Container Platform version manually until the support is included in IBM Storage Fusion.

For the upgrade procedure, Red Hat OpenShift Container Platform, see [Understanding OpenShift Container Platform updates](#).

- [Prerequisites for enterprise registry upgrade](#)

If you installed the earlier version of IBM Storage Fusion by using your enterprise registry, then mirror the 2.8.0 images to your enterprise registry.

- [Upgrading IBM Storage Fusion services](#)

From the IBM Storage Fusion user interface, upgrade the IBM Storage Fusion services, namely Data Foundation, Global Data Platform, Backup & Restore, and Data Cataloging.

- [Upgrading Red Hat OpenShift Data Foundation 4.12 to IBM Storage Fusion Data Foundation 4.12 or higher](#)

Steps to upgrade Red Hat Data Foundation 4.12 to IBM Storage Fusion Data Foundation 4.12 or higher.

Prerequisites for enterprise registry upgrade

If you installed the earlier version of IBM Storage Fusion by using your enterprise registry, then mirror the 2.8.0 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. Mirror IBM Storage Fusion images. See [Mirroring IBM Storage Fusion images](#).
3. Mirror Backup & Restore images. See [Mirroring Backup & Restore images](#).
4. Mirror IBM Storage Scale images. See [Mirroring IBM Storage Scale images](#).
5. Data Cataloging offline upgrade:

- a. Complete steps [2 to 7 of Mirroring Data Cataloging images](#) procedure.
- b. 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
```

- c. 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
```

6. IBM Fusion Data Foundation offline service upgrade:

Offline upgrade of Data Foundation images that are deployed on OpenShift® Container Platform version 4.12, 4.13, 4.14, or 4.15:

- a. Complete steps [1 to 7 of Mirroring Data Foundation images deployed on OpenShift Container Platform version 4.14 or 4.15](#) topic.
- b. Before you upgrade IBM Storage Fusion, from the Services page of the IBM Storage Fusion user interface, disable Automatic updates for Data Foundation service.
- c. Go to Operators > Installed Operators > IBM Storage Fusion Data Foundation > Subscription, and check whether the Update approval is changed to Manually.
- d. Start the IBM Storage Fusion version upgrade.
- e. Update the image digest ID after you upgrade the IBM Storage 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 Storage Fusion user interface.
7. 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
    # source 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 Storage Fusion services

From the IBM Storage Fusion user interface, upgrade the IBM Storage Fusion services, namely Data Foundation, Global Data Platform, Backup & Restore, and Data Cataloging.

Before you begin

Important: It is recommended to upgrade the services to the latest version after a IBM Storage Fusion upgrade to avoid compatibility issues between IBM Storage Fusion and its installed services.

About this task

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 Storage Fusion user interface:
 - Check the notification banner on the Quick start page.
 - Click the bell icon and view notifications in System notifications.
 - Check the notification banner on the Services page.

Pre-checks

When you initiate upgrade, the Upgrade <service name> window gets displayed. The IBM Storage Fusion HCI System runs common pre-checks on the OpenShiftLifeCycleManagementPodStatus, CatalogSourceStatus, FusionOperatorStatus, DNSResolution, and RegistryAccessibility. Wait for a moment to see the precheck status in the IBM Storage Fusion user interface.

If all the checks pass or there are some conditions in warning state, you can proceed with the upgrade. Whenever critical errors are found in prechecks, you cannot proceed with upgrade instead you must fix the issue to continue with upgrade. To know more about critical errors or warnings, click View report link. The IBM Storage Fusion HCI System user interface takes you to the Services page wherein a Upgrade precheck report window displays all the Must fix and Optional warnings. You can click on the corresponding BMYxxxxx 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.

Upgrade Data Foundation service

Before you begin

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 4.15 is available for Data Foundation service. Do the following steps to manually upgrade the service:

Procedure

1. Log in to IBM Storage Fusion user interface.
2. Click Services menu to go to the Services page.
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 Storage Fusion runs common pre-checks during upgrade. You can easily 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. To know more about the pre-checks, see [Pre-checks in the About the tasks section](#).
4. Wait for a minute or more to see the upgrade status in the IBM Storage 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.

Upgrade Global Data Platform

Before you begin

Run the `mmhealth` command to check the scale common health status of the storage scale cluster.

About this task

If you want to continue to use Global Data Platform 5.2.0.0, then the OpenShift Container Platform version must be higher than 4.13.

If you want to continue to use Global Data Platform 5.1.9.x, then you can upgrade IBM Storage Fusion to 2.8.0 but must not upgrade OpenShift Container Platform from version 4.12. Ensure you do not upgrade Global Data Platform without upgrading OpenShift Container Platform first.

If you want to upgrade to Global Data Platform 5.2.0.0 for IBM Storage Fusion 2.8.0, do the following steps:

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.
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 Storage Fusion runs common pre-checks during upgrade. You can easily 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. To know more about the pre-checks, see [Pre-checks in the About the tasks section](#).
Note: Wait for a minute or more to see the upgrade status in the IBM Storage 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 the IBM Storage Scale 5.2.0. For more information, see [Remove deprecated resources](#).


```
oc delete ClusterRoleBinding ibm-spectrum-scale-dns
oc delete ClusterRole ibm-spectrum-scale-dns
```

Upgrade Backup & Restore

Follow this procedure to upgrade the Backup & Restore version.

About this task

- If the previous version of Backup & Restore service has only Hub and you want Spoke as well post upgrade, then you need to do a Spoke installation.
- When you upgrade the Hub to 2.8, the AMQ streams operator does not get upgraded to the version that is normally installed by a new 2.8 installation. It is because at least one spoke is at a version less than 2.8. This down-level version of AMQ remains on the hub until, at least, you do the next 2.8 upgrade on the Hub.
- When you add new Spokes, ensure that they are at the same upgraded version level as the Hub.

Procedure

1. Log in to IBM Storage Fusion user interface.
2. Click Services menu to go to the Services page.
3. In the Installed section, click Upgrade link for the Backup & Restore service. Alternatively, from the  (ellipsis overflow menu) of Backup & Restore, click Upgrade.
You can see the Upgrade option only when an upgrade is available for the service.
The IBM Storage Fusion runs common pre-checks during upgrade. You can easily 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. To know more about the pre-checks, see [Pre-checks in the About the tasks section](#). After the completion of upgrade, a success notification gets displayed.

Upgrade Data cataloging

About this task

Sometimes, the Data Cataloging service upgrade may not be available after you upgrade IBM Storage Fusion. To know more about such issues and its resolutions, see [Troubleshooting Data Cataloging service upgrade issues](#).

Important: You cannot enable Data Cataloging service in an upgraded IBM Storage Fusion 2.8 with Red Hat® OpenShift Container Platform 4.15.

Procedure

Upgrade **Data Cataloging** service as follows:

- a. In the Installed section, click Upgrade in the ellipsis overflow menu of Data Cataloging.
- b. In the Upgrade service window, click **Upgrade**.

The upgrade starts and you can see the upgrade status. The IBM Storage Fusion runs common pre-checks during upgrade. You can easily 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.

To know more about the pre-checks, see [Pre-checks in the About the tasks section](#). 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 Storage Fusion user interface.

Upgrading Red Hat OpenShift Data Foundation 4.12 to IBM Storage Fusion Data Foundation 4.12 or higher

Steps to upgrade Red Hat® Data Foundation 4.12 to IBM Storage Fusion Data Foundation 4.12 or higher.

Before you begin

- Ensure that the OpenShift® Data Foundation version installed is ≥ 4.12 .
- Ensure that the OpenShift Data Foundation cluster is healthy and data is resilient. You can get the details from IBM Storage 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 Storage Fusion Data Foundation catalogsource exists:
 1. Verify the `isf-data-foundation-catalog` is created. The IBM Storage 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

About this task

Note:

- If you installed the latest version of OpenShift Data Foundation that is at a higher level than the latest IBM Storage Fusion Data Foundation version, then upgrade to IBM Storage Fusion Data Foundation is not possible.
- This procedure is also applicable for versions 4.12 or higher. The supported versions for IBM Storage Fusion 2.7 is 4.12 and 4.13.

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.coreos.com/odf-operator --patch '{"spec":{"source":"isf-data-foundation-catalog"}}' --type=merge
```

- b. Update the mcg-operator/ocs-operator/odf-csi-addons-operator subscriptions' catalogsource name.

```
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. Make sure all the catalogsource name is updated.

```
oc get subscriptions.coreos.com -n openshift-storage
```

Sample output:

NAME	PACKAGE	SOURCE
mcg-operator-stable-4.13-redhat-operators-openshift-marketplace	mcg-operator	isf-data-foundatio
ocs-operator-stable-4.13-redhat-operators-openshift-marketplace	ocs-operator	isf-data-foundatio
odf-csi-addons-operator-stable-4.13-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 Storage 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 Storage 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.v4.13	NooBaa Operator	4.13	mcg-operator.v4.13-rhodf	Succeeded
ocs-operator.v4.13	Container Storage	4.13	ocs-operator.v4.13-rhodf	Succeeded
odf-csi-addons-operator.v4.13	CSI Addons	4.13	odf-csi-addons-operator.v4.13-rhodf	Succeeded
odf-operator.v4.13	IBM Storage Fusion Data Foundation	4.13	odf-operator.v4.13-rhodf	Succeeded

Note: If any of the verification steps fail, contact IBM Support.

Deploying your workloads

Deployment of your workloads with Persistent volumes on IBM Storage Fusion.

The managing workloads topic deals with anything that is related to workloads on IBM Storage Fusion.

- [Deploying your workloads with Persistent volumes](#)

This section shows you how you can provision Persistent volumes for application data persistence on IBM Storage Fusion .

Deploying your workloads with Persistent volumes

This section shows you how you can provision Persistent volumes for application data persistence on IBM Storage Fusion .

A PersistentVolume (PV) is a representation of storage volume in the IBM Storage 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.

PersistentVolumeClaims allows you to 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 details 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 details on creating dynamic provisioning, see [Dynamic provisioning of Persistent Volume in filesystem](#).

Storage classes

The storage class options for IBM Storage 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.

- [Storage provisioning using Container Storage Interface driver](#)

This section discusses the automatic storage provisioning in IBM Storage Fusion. Also, it provides references to storage class options for IBM Storage Fusion by using Container Storage Interface driver.

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-mcgfs3-14544310586575975773/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-mcgfs3-14544310586575975773/images`.

2. Run the following apply command to create a Persistent Volume:

```
kubectl apply -f pv.yaml
```

Note:

- IBM Storage Fusion creates static Persistent Volume templates for each filesystem.
- For IBM Storage 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 Storage Fusion 2.7.1 or higher supports IBM Storage Scale 5.1.9. IBM Storage 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 [#sf_sds_storage_ex1](#) and [sf_sds_storage.html#sf_sds_storage_ex2](#). 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 <https://www.ibm.com/docs/en/storage-scale-csi?topic=driver-volume-expansion>.

The IBM Storage 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 Storage 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 Storage 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-mcgfs3-14544310586575975773"
  filesetType: "dependent"
  parentFileset: "primary-fileset-ibm-spectrum-scale-fs-mcgfs3-14544310586575975773-2822920183822285462"
  reclaimPolicy: Delete
  allowVolumeExpansion: true
```

The file set path pattern is **primary-fileset-\$FS_CR_NAME-\$LocalClusterID**. If the fileset does not exist, then IBM Storage 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 Storage Fusion:

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	A 2
ibm-storage-fusion-cp-sc	spectrumscale.csi.ibm.com	Delete	Immediate	true	

```
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 ReadWriteOnce (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 ReadWriteOnce (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 Storage Fusion](#).

Storage provisioning using Container Storage Interface driver

This section discusses the automatic storage provisioning in IBM Storage Fusion. Also, it provides references to storage class options for IBM Storage Fusion by using Container Storage Interface driver.

Note: IBM Storage Fusion 2.7.1 or higher supports IBM Storage Scale 5.1.9. IBM Storage Fusion 2.7 supports IBM Storage Scale 5.1.7.

For IBM Storage Fusion, storage provisioning is automated through the Container Storage Interface driver. IBM Storage Fusion provides two choices for storage provisioning:

IBM Fusion Data Foundation

IBM Storage Fusion provides entitlement for IBM Fusion Data Foundation.

Remote IBM Storage Scale file system mount

Connect to an external IBM Storage Scale cluster to access an existing file system. Container workloads that run on OpenShift® can access existing data sets stored in the IBM Storage Scale file system and provision new persistent volumes backed by the storage from the remote IBM Storage Scale files system.

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 [Storage class in IBM Storage Scale Container Storage Interface Driver documentation](#).

For more information about IBM Storage Scale Container Native, see [Introduction to IBM Storage Scale Container Native](#). For more information about how you can provision Kubernetes persistent volumes from IBM Storage Scale ECE by using Container Network Interface (CNI), see [IBM Storage Scale Container Storage Interface Driver documentation](#).

Configuring Data Foundation storage

From the Data Foundation user interface page, you can configure IBM Fusion Data Foundation service.

After the configuration is complete, you can view capacity utilization and storage nodes. You can add nodes to your storage configuration and scale its capacities.

Support is also available to configure IBM Fusion Data Foundation encryption from the IBM Storage Fusion user interface.

- [**Configuring Data Foundation local storage**](#)

From the Data Foundation user interface page, you can configure IBM Fusion Data Foundation storage.

- [**Configuring Data Foundation dynamic storage**](#)

From the Data Foundation user interface page, you can configure dynamic storage for IBM Fusion Data Foundation.

- [**Configuring Data Foundation in external mode**](#)

From the Data Foundation user interface page, you get redirected to OpenShift® console to create an external storage system.

Configuring Data Foundation local storage

From the Data Foundation user interface page, you can configure IBM Fusion Data Foundation storage.

About this task

Install service with device type as local.

Procedure

1. Go to the Data Foundation page.
2. Wait for the discovery of compute nodes to complete.
After the discovery process completes, the Configure storage button is enabled.
3. Click Configure storage.
4. If you want to change the default disk size for your storage configuration, select from the available disk size and type in the Configure storage page. The type can be HDD or SSD/NVMe. If you have either SSD or HDD types available, then this option is not available for selection. The SSD/NVMe is the recommended option for optimal performance and reliability. The HDDs are supported for development or test only.
5. In the Storage nodes section, select nodes from the node table based on the recommendation. They are candidate nodes to be selected as IBM Fusion Data Foundation storage nodes. Only compute nodes with available HDD or SSD/NVMe disks can be viewed as candidate nodes and they get displayed in the table. The table includes Name, Disks, Disk count, Disk size (TiB), CPU core, and Memory (GiB) details about the node.
For example, the recommendation can be to select a minimum of 3 nodes with an aggregation of at least 30 CPUs and 72 GiB memory.
6. In the Summary section, check the capacity configuration and click Next.
7. In the Additional settings section, you can toggle to enable dedicated nodes for infrastructure.
These nodes get tainted to only allow IBM Fusion Data Foundation workload to be deployed on them. Also, these dedicated nodes for infrastructure option changes the selected compute nodes to infrastructure nodes, and you can save subscription costs of OpenShift® Container Platform for these Data Foundation nodes. When the Global Data Platform service is enabled, the Dedicated nodes for infrastructure button is not visible.
8. In the Enable encryption section, enter the following details:
 - Store the encryption key as a secret in the cluster.
 - Store the encryption key in an external KMS:If you select Store the encryption key in an external KMS option, then enter the following connection settings:
 - Enter the Hostname/ IP address of your KMS server.
 - Enter the value of Port of your KMS server.
 - Select a Provider type. It can be Vault or Thales CipherTrust Manager.

Table 1. Provider type options

Provider type	Procedure
Vault	<p>For Vault, enter the following details.</p> <ul style="list-style-type: none">• Select an Authentication method. It can be Token or Kubernetes. <p>Token method</p> <p>If you select the method as Token, then enter value for token. For more information on how to create token in vault server, see Enabling encryption with the token authentication using HashiCorp Vault(manual part) in Preparing to connect to an external KMS server in IBM Fusion Data Foundation.</p> <p>Kubernetes method</p> <p>If you select the method as Kubernetes, then enter value for role. After you click Configure in the next step, manually do the steps that are defined in the Enabling encryption with the Kubernetes authentication using HashiCorp Vault (manual part). The role will be generated with <code>rook-ceph-system</code>, <code>rook-ceph-osd</code>, <code>noobaa</code> as <code>bound_service_account_names</code> in the Vault by the manual steps.</p> <ul style="list-style-type: none">• Enter the Backend path that you defined in step 1 in Enabling encryption with the token authentication using HashiCorp Vault(manual part) or step 3.b that defined in Enabling encryption with the Kubernetes authentication using HashiCorp Vault (manual part).• Optionally, enter the CA certificate, Client certificate, Client private key (optional) in pem format. Note: Client certificate and client private key need to be provided as a pair, or neither of them. Only providing one of them is invalid.• Optionally, enter TLS server name• Optionally, if authentication method is Token, enter the Vault enterprise namespace.• Optionally, if authentication method is Kubernetes, enter Authentication path. <p>To get more information about each of these fields, see Enabling encryption with the token authentication using HashiCorp Vault(manual part). For more information about TLS server name, Vault enterprise namespace, and Authentication path. See https://developer.hashicorp.com/vault/docs.</p>
Thales CipherTrust Manager	<p>For Thales CipherTrust Manager, enter the following details:</p> <ul style="list-style-type: none">• Enter the CA certificate generated in step 6 in Enabling encryption using Thales CipherTrust Manager (manual part).• Enter the Client certificate and the private key generated in step 4 of Enabling encryption using Thales CipherTrust Manager (manual part).• Optionally, enter TLS server name.

9. Click Configure.

The Data Foundation page now includes Usable capacity, Health, and Storage nodes sections:

Important: If you use an external device, then the Storage nodes list is not available instead the following message is displayed:

External mode Data Foundation is deployed in external mode.

Note: Sometimes, it can take up to five minutes to show the summary of Usable capacity and Health sections.

Usable capacity

The amount of capacity that is available for storing data on a system after the RAID or mirroring technology is applied. In IBM Fusion Data Foundation, it is 1/3 of the raw capacity when you use three replicas. Usable capacity is represented in a line graph. The block, file and object are distinguished by different

colors.

Health

It includes Storage cluster and Data resiliency. The status gets displayed only after the provisioning is complete. You can check in the Storage > Data Foundation. Go to Storage Systems to see the created storage file system. Open the file system and in the Overview tab. In the Storage > Persistent Volumes page, you can view the local persistent volumes based on the selected disk.

Storage nodes

It includes a list of all nodes that are used in your local storage configuration. The node details listed in the table are Name, Status, Disks, Disk size (TiB), CPU, and Memory (GiB). You can use the search option to filter and search for nodes. You can add nodes to scale up. For the procedure to add nodes, see [Adding nodes to your IBM Fusion Data Foundation storage](#).

What to do next

- For Kubernetes method, finish the manual steps that are defined in [Enable encryption with KMS using the Kubernetes authentication method](#).
- You can now add nodes and disks.
- If you have encryption settings, you can edit the details.
- **[Adding disks to storage nodes](#)**
You can add disks to local storage nodes to expand IBM Fusion Data Foundation capacity. IBM Storage Fusion auto detects the added local device and auto expands the capacity to IBM Fusion Data Foundation storage cluster.
- **[Adding nodes to your IBM Fusion Data Foundation storage](#)**
You can select additional nodes to expand your local storage configuration.
- **[Configuring local stretch cluster DR with Fusion Data Foundation](#)**
Steps to configure disaster recovery with stretch cluster for Fusion Data Foundation.

Adding disks to storage nodes

You can add disks to local storage nodes to expand IBM Fusion Data Foundation capacity. IBM Storage Fusion auto detects the added local device and auto expands the capacity to IBM Fusion Data Foundation storage cluster.

About this task

If you deployed with less than three zones enabling the flexible scale deployment, add any number of disks to scale up. Unlike Internal (Dynamic Provisioning) mode deployment, there is no constraint to scale up by three disks at a time.

Procedure

1. Ensure that the devices to be added to the storage nodes are the same size and type as the existing IBM Fusion Data Foundation devices.
2. Run the following command to check IBM Fusion Data Foundation replica number:

```
oc get storagecluster -n openshift-storage -o yaml | grep replica
```

If replica is 1, it means that IBM Fusion Data Foundation flexible scaling is auto enabled in a OpenShift® Container Platform with less than 3 zones. In flexible scaling mode, you could add any number of disks to any storage node. But the recommendation is to add disk number in multiples of 3 to node number in multiple of 3. These added disks must be evenly distributed in these nodes.

If replica is 3, this means that IBM Fusion Data Foundation is not in flexible scaling mode. In such case, you must add disk number in multiple of 3 to storage node number in multiple of 3. Also, these disks and nodes must be evenly distributed in different zones.

3. Add disks to storage nodes and wait for IBM Fusion Data Foundation capacity to be auto scaled up.

Adding nodes to your IBM Fusion Data Foundation storage

You can select additional nodes to expand your local storage configuration.

Before you begin

Configure IBM Fusion Data Foundation local storage configuration. For more information about the configuration, see [Configuring Data Foundation local storage](#).

About this task

Note: When the IBM Fusion Data Foundation is in dedicated mode and Global Data Platform service is also enabled, you cannot add nodes to scale out. For Red Hat® OpenShift® Data Foundation deployed outside of IBM Storage Fusion, you cannot add nodes from IBM Storage Fusion instead you get redirected to add nodes from OpenShift Container Platform web console.

Procedure

1. In the Data Foundation page, click Add nodes in the Storage nodes section.
The Add storage nodes window gets displayed.
2. In the Add storage nodes window, select one or more nodes.

The Summary section includes usable capacity, current total, and projected total. The Current total indicates the previous selection and Projected total is an aggregation of the previous and new selection.

3. Click Add.

The Data Foundation page is displayed. After the selected node is added to the configuration successfully, the Add button changes to Configuring. After the configuration completes, the button changes to Complete with a green tick mark. When there are no more nodes available for expansion, the Add nodes button gets grayed out.

Configuring local stretch cluster DR with Fusion Data Foundation

Steps to configure disaster recovery with stretch cluster for Fusion Data Foundation.

Before you begin

- OpenShift® Container Platform cluster version must be 4.13 or higher.
 - Ensure that you have at least three OpenShift Container Platform control nodes in three different zones. One master node in each of the three zones.
 - Ensure that you have at least four OpenShift Container Platform compute nodes evenly distributed across the two Data Zones.
- The stretch cluster solution is designed for deployments where latencies do not exceed 5 ms between zones, with a maximum round-trip time (RTT) of 10 ms.
- Both flexible scaling and arbiter cannot be enabled at the same time as they have a conflicting scaling logic. With Flexible scaling, you can add one node at a time to your Red Hat® OpenShift Data Foundation cluster. Whereas, in an arbiter cluster, you need to add at least one node in each of the two data zones.
- Ensure that each node is pre-labeled with its zone label. 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
```

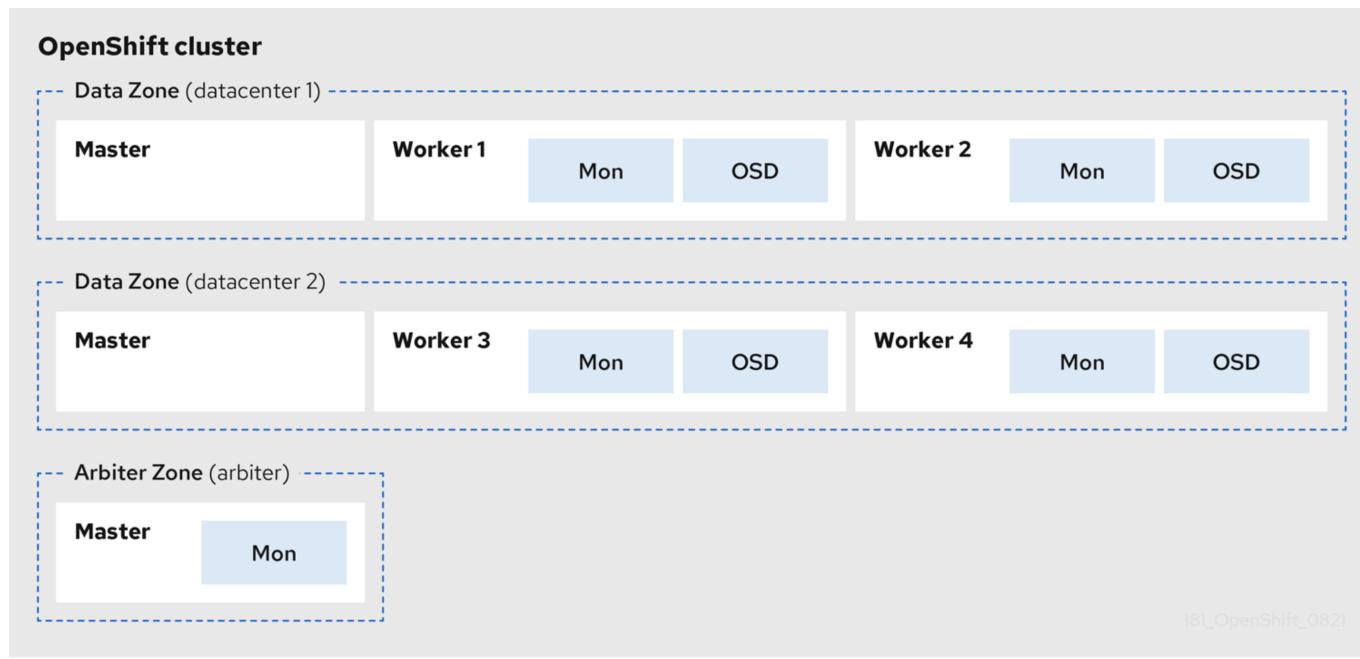
To apply the labels to the node:

```
oc label node <NODENAME> topology.kubernetes.io/zone=<LABEL>
```

About this task

If you want to deploy stretch cluster with RH Red Hat OpenShift Data Foundation, see [Disaster recovery with stretch cluster for Red Hat OpenShift Data Foundation](#).

The following diagram shows the simplest deployment for a stretched cluster:



Procedure

- Go to Services page and enable Fusion Data Foundation with local mode.
- Create Storage stretch cluster.

Note: If you wish to use the Arbiter stretch cluster, a minimum of 4 nodes (2 different zones, 2 nodes per zone) and 1 additional zone with 1 node is required. All nodes must be pre-labeled with zones to be validated on cluster creation.

For Red Hat OpenShift Data Foundation documentation, see [Creating OpenShift Data Foundation cluster](#).

- Create storageCluster from the OpenShift Container Platform console.
 - Go to Operators > Create StorageSystem of `openshift-storage` namespace.
 - In the Backing storage tab page, select local storage devices and click Next.
 - In the Create local volume set, enter LocalVolumeSet name and StorageClass Name.
 - From the node details, ensure there are at least two nodes in one zone and at 2 zones.

- v. In the Create LocalVolumeSet window, click Yes to confirm.
After the LocalVolumeSet is created you cannot edit it. If you wish to use the Arbiter stretch cluster, a minimum of 4 nodes (two different zone, two nodes per zone) and one additional zone with one node is required. All nodes must be per-labeled with zones to be validated on cluster creation.
 - vi. Click Next.
 - vii. In the Capacity and nodes tab, select the Arbiter zone from the drop down list.
 - viii. Click Next.
 - ix. In the Security and network tab, click Next.
 - x. In the Review and create tab, click Create StorageSystem.
3. Validate storage stretch cluster from OpenShift Container Platform console.
- a. Log in to OpenShift Container Platform console.
 - b. Go to Storage > Data Foundation.
The Overview tab is displayed by default.
 - c. Go to Topology tab and view the details.
4. Validate storage stretch cluster from IBM Storage Fusion user interface:
- a. Log in to IBM Storage Fusion user interface.
 - b. In the Data Foundation page, view the storage nodes.

Configuring Data Foundation dynamic storage

From the Data Foundation user interface page, you can configure dynamic storage for IBM Fusion Data Foundation.

Before you begin

Install IBM Fusion Data Foundation service with device type as dynamic.

Procedure

1. Go to the Data Foundation page.
 2. Click Configure storage.
IBM Storage Fusion automatically detects the device type as dynamic storage.
 3. In the Configure storage page, select the following values in the Storage capacity:
 - Disk size (TiB)
Select from the available storage capacities listed in the Disk size (TiB) drop down menu.
 - Cloud volume storage class
Select a storage class value.
 - Usable capacity (TiB)
After you select a value in Disk size (TiB), the Usable capacity is enabled. Click the + or - to select a value that is increments of the disk drive.
Nodes listed in the node table are candidate nodes to be selected as IBM Fusion Data Foundation storage nodes.
 4. In the Select storage nodes section, select the nodes based on the recommendation displayed on the page.
You can also use the Data Foundation sizer tool to come up with your calculations. The table includes Name, Zone, CPU, and Memory (GiB) details about the node. For example, the recommendation can be select a minimum of 3 nodes with an aggregation of at least 30 CPUs and 72 GiB memory. Additionally, a message also prompts you to select nodes from each of the available zones.
 5. In the Additional settings section, you can toggle to enable dedicated nodes for infrastructure.
These nodes get tainted to only allow IBM Fusion Data Foundation workload to be deployed on them. Also, these dedicated nodes for infrastructure option changes the selected compute nodes to infrastructure nodes, and you can save subscription costs of OpenShift® Container Platform for these Data Foundation nodes. When the Global Data Platform service is enabled, the Dedicated nodes for infrastructure button is not visible.
 6. In the Summary section, check the capacity configuration.
Note: Sometimes, it can take up to five minutes to show the summary of Usable capacity and Health sections.
- Usable capacity**
The amount of capacity that is available for storing data on a system after the RAID or mirroring technology is applied. In IBM Fusion Data Foundation, it is 1/3 of the raw capacity when you use three replicas. Usable capacity is represented in a line graph. The block, file and object are distinguished by different colors.
- Health**
It includes Storage cluster and Data resiliency. The status gets displayed only after the provisioning is complete. You can check in the Storage > Data Foundation. Go to Storage Systems to see the created storage file system. Open the file system and in the Overview tab. In the Storage > Persistent Volumes page, you can view the local persistent volumes based on the selected disk.
- Storage nodes**
It includes the list of all nodes used in your local storage configuration. The node details listed in the table are Name, Status, Disks, Disk size (TiB), CPU, and Memory (GiB). You can use the search option to filter and search for nodes. You can add nodes to scale up. For the procedure to add nodes, see [Adding nodes to your IBM Fusion Data Foundation storage](#)
7. Click Next.
The Enable encryption section is displayed.
 8. In the Enable encryption section, enter the following details:
 - Store the encryption key as a secret in the cluster.
 - Store the encryption key in an external KMS:
If you select Store the encryption key in an external KMS option, then enter the following connection settings:
 - Enter the Hostname/ IP address of your KMS server.
 - Enter the value of Port of your KMS server.
 - Select a Provider type. It can be Vault or Thales CipherTrust Manager.

Table 1. Provider type options

Provider type	Procedure
Vault	<p>For Vault, enter the following details.</p> <ul style="list-style-type: none"> Select an Authentication method. It can be Token or Kubernetes. <p>Token method</p> <p>If you select the method as Token, then enter value for token. For more information on how to create token in vault server, see Enabling encryption with the token authentication using HashiCorp Vault(manual part) in Preparing to connect to an external KMS server in IBM Fusion Data Foundation.</p> <p>Kubernetes method</p> <p>If you select the method as Kubernetes, then enter value for role. After you click Configure in the next step, manually do the steps that are defined in the Enabling encryption with the Kubernetes authentication using HashiCorp Vault (manual part). The role will be generated with <code>rook-ceph-system</code>, <code>rook-ceph-osd</code>, <code>noobaa</code> as <code>bound_service_account_names</code> in the Vault by the manual steps.</p> <ul style="list-style-type: none"> Enter the Backend path that you defined in step 1 in Enabling encryption with the token authentication using HashiCorp Vault(manual part) or step 3.b that defined in Enabling encryption with the Kubernetes authentication using HashiCorp Vault (manual part). Optionally, enter the CA certificate, Client certificate, Client private key (optional) in pem format. Note: Client certificate and client private key need to be provided as a pair, or neither of them. Only providing one of them is invalid. Optionally, enter TLS server name Optionally, if authentication method is Token, enter the Vault enterprise namespace. Optionally, if authentication method is Kubernetes, enter Authentication path. <p>To get more information about each of these fields, see Enabling encryption with the token authentication using HashiCorp Vault(manual part). For more information about TLS server name, Vault enterprise namespace, and Authentication path. See https://developer.hashicorp.com/vault/docs.</p>
Thales CipherTrust Manager	<p>For Thales CipherTrust Manager, enter the following details:</p> <ul style="list-style-type: none"> Enter the CA certificate generated in step 6 in Enabling encryption using Thales CipherTrust Manager (manual part). Enter the Client certificate and the private key generated in step 4 of Enabling encryption using Thales CipherTrust Manager (manual part). Optionally, enter TLS server name.

9. Click Configure.

- The Data Foundation page is displayed. After the configuration is complete, the Data Foundation loads the Usable capacity, Health, and Storage nodes.
- If Kubernetes authentication method is used in an external KMS, installation will be pending until the user finishes the manual steps in the Vault server.

What to do next

- For Kubernetes method, finish the manual steps that are defined in [Enable encryption with KMS using the Kubernetes authentication method](#).
- You can now add nodes and capacity.
- If you have encryption settings, you can edit the details.
- [Adding capacity to storage nodes](#)**
You can add disks to storage nodes to expand IBM Fusion Data Foundation capacity.
- [Adding nodes to your IBM Fusion Data Foundation storage](#)**
You can select additional nodes to expand your dynamic storage configuration.

Adding capacity to storage nodes

You can add disks to storage nodes to expand IBM Fusion Data Foundation capacity.

About this task

If Red Hat® OpenShift® Data Foundation is deployed outside of IBM Storage Fusion, then you cannot add capacity from IBM Storage Fusion.

Procedure

- In the Data Foundation page, click Add capacity in the Storage nodes section.
The Add capacity window gets displayed.
- In the Add capacity window, enter the usable capacity.
The value must be in increments of drive size.
- Click Add.
A notification is displayed indicating that the capacity addition is in progress.

Adding nodes to your IBM Fusion Data Foundation storage

You can select additional nodes to expand your dynamic storage configuration.

Before you begin

You must have completed the IBM Fusion Data Foundation dynamic storage configuration. For more information about the configuration, see [Configuring Data Foundation dynamic storage](#).

About this task

Note: When the IBM Fusion Data Foundation is in dedicated mode and Global Data Platform service is also enabled, you cannot add nodes to scale out. For Red Hat® OpenShift® Data Foundation deployed outside of IBM Storage Fusion, you cannot add nodes from IBM Storage Fusion.

Procedure

1. In the Data Foundation page, click Add nodes in the Storage nodes section.
The Add storage nodes window gets displayed.
2. In the Add storage nodes window, select one or more nodes.
The Storage nodes table includes Name, Zone, CPU, and Memory details. The Summary section includes total number of nodes, CPUs, memory, Zones, current total, and projected total. The Current total indicates the previous selection and Projected total is an aggregation of the previous and new selection.
3. Click Add.
The Add button changes to Adding nodes. After the selected nodes is added to the configuration successfully, then the Data Foundation page is displayed with the newly added node(s) in Ready state. When there are no more nodes available for expansion, the Add nodes button gets grayed out.

Configuring Data Foundation in external mode

From the Data Foundation user interface page, you get redirected to OpenShift® console to create an external storage system.

Before you begin

- Install Fusion Data Foundation service with device type as external.
- IBM Storage Ceph must have Ceph Dashboard that is installed and configured. For more information, see [Dashboard > Ceph Dashboard installation and access of IBM Storage Ceph documentation](#).
- It is recommended that the external IBM Storage Ceph cluster has the PG Autoscaler enabled.
- The external Ceph cluster must have an existing RBD pool pre-configured for use. If it does not exist, contact your IBM Storage Ceph administrator to create one before you move ahead with Fusion Data Foundation deployment. IBM recommends to use a separate pool for each Fusion Data Foundation cluster.
- Optional: If there is a zonegroup created apart from the default zonegroup, you need to add the hostname, `rook-ceph-rgw-ocs-external-storagecluster-cephobjectstore.openshift-storage.svc` to the zone group, as Fusion Data Foundation sends S3 requests to the RADOS Object Gateways (RGWs) with this hostname.

Procedure

1. Create a StorageSystem.
 - a. Go to the Data Foundation page.
 - b. Click Configure storage.
 - c. Click Create StorageSystem in the newly opened OpenShift Container Platform console page.
 - d. 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.
 - e. Click Next.
 - f. In the Connection details page, provide the necessary information:
 - i. Click on 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, 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.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>"}}]
```

3. 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.

4. 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.

- g. 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.

- h. Click Create StorageSystem.

2. Verify the StorageSystem creation.

- a. From the OpenShift web console, go to Installed Operators > IBM Storage Fusion Data Foundation > Storage System > ocs-external-storagecluster-storagesystem > Resources.

- b. Verify that StorageCluster is in a Ready state and has a green tick.

3. Verify Data Foundation page.

From IBM Storage Fusion user interface, go to the Data Foundation page. It shows that the Data Foundation is deployed in external mode, and also shows Usable capacity and Health info in the page.

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 Storage Fusion to access an encrypted remote IBM Storage Scale file system.

- [Preparing to connect to an external KMS server in IBM Fusion Data Foundation](#)

Procedure to prepare for the connection to an external KMS from IBM Fusion Data Foundation.

Configuring encryption for Global Data Platform storage

You can configure the encryption in IBM Storage Fusion to access an encrypted remote IBM Storage Scale file system.

Before you begin

Prepare IBM® Security Guardium® Key Lifecycle Manager (GKLM) server for IBM Storage 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](#).

To know more 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 Storage 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 Storage 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 Storage 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 GKLM](#).
- User name
The administrator user name for GKLM Server. The default value is `GKLMAadmin`.
- 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 Storage 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 Storage Fusion system must use the same RKMID, which describes a combination of keyserver, tenant, and client on the remote scale cluster.
- 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 details 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.

Preparing to connect to an external KMS server in IBM Fusion Data Foundation

Procedure to prepare for the connection to an external KMS from IBM Fusion Data Foundation.

Before you begin

- For external Key Management System (KMS), choose either HashiCorp Vault or Thales CipherTrust Manager.
- You must install IBM Storage 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 Data Foundation local storage](#) and [Configuring Data Foundation dynamic storage](#).

IBM 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 IBM 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 [IBM Fusion Data Foundation service error scenarios](#).

For additional reference, go through the following documents:

- [Red Hat Data Foundation 4.14 Data encryption options](#)
- [Red Hat Data Foundation 4.15 Data encryption options](#)
- [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": "IFNbUowWAzg5cyiUZGv1vf5o",
    "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

- Get the **VAULT_SA_SECRET_NAME** in OpenShift® Container Platform:

a. For OpenShift Container Platform 4.10, 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 4.12, assign a default value for 'VAULT_SA_SECRET_NAME'"

```
VAULT_SA_SECRET_NAME=odf-vault-auth-token
```

- 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
```

- 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 will use 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
```

```
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 Storage 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 encryption using Thales CipherTrust Manager (manual part)

Configure the Key Management Interoperability Protocol (KMIP) settings in Thales CipherTrust Manager server.

About this task

IBM Storage Fusion HCI System 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 Storage Fusion HCI System 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 Storage Fusion user interface, use these downloaded files.

Accessing remote IBM Storage Scale storage cluster in IBM Storage Fusion

You can configure a remote file system mount for IBM Storage Scale and access it. The container native data access is only for IBM Storage Fusion on On-premises, Bare Metal, VMware, and Amazon Web Services deployments.

Before you begin

- 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 remotely 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 Storage 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

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 Storage Fusion 2.8.0)
- Standalone rack with FDF (upgraded to IBM Storage Fusion 2.8.0 / newly installed with IBM Storage Fusion 2.8.0)

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.

1. The remote file system mount must be configured. For more information, see [Connecting to remote IBM Storage Scale file systems](#).
2. When the remote file system mount is created, the IBM Storage Fusion automatically creates a statically provisioned volume that points to the file system. You must know how to identify this Persistent Volume.
3. 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.
4. 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.
5. 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 Storage 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 Storage Fusion, from a containerized workload, you can access and store data from existing IBM Storage Scale clusters.

Before you begin

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 Storage 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 Storage Fusion. You can check whether the IBM Storage 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 Storage Fusion user interface. For the procedure, see [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 Storage 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
```

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
```
 - 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 gpfgui service is running:

```
systemctl status gpfgui
```

If gui service has issues, stop and start the service:

```
systemctl stop gpfgui.service
systemctl start gpfgui.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

- Click Storage > Remote file systems menu in the IBM Storage Fusion user interface.
- Click Add file system.
The Add IBM Spectrum Scale file system slide out pane gets displayed.
- 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 Storage Fusion verifies the certificate chain and hostname of the IBM Storage Scale cluster. If you do not provide the root CA value, the IBM Storage 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 Storage 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 Storage 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.

Applications

Applications contain resources, such as microservices, databases, and other stateful data distributed across projects. IBM Storage 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	<p>The Applications page lists all your OpenShift applications with their backup details, such as Name, Used, Capacity, Backup status, Last backup on, and Policies.</p> <p>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.</p>
Actions	<p>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.</p> <p>Assign backup policy To assign backup policy, see Assign backup policy.</p> <p>Backup now To start the backup of an application right away, click Backup now in the ellipsis menu of the application record.</p> <p>Restore Restore the application backup. For the steps to restore an application resource, see Restoring an application.</p>

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.

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 from Applications list page](#).

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 from Applications list page](#).

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 utilizing 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.
Disaster recovery This section is available only in a Metro-DR setup.	The Disaster recovery section displays the disaster recovery to which applications are enrolled. It shows Regional status, Metro status, Current site, Home site, and Last consistent time.

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: <ul style="list-style-type: none">• 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. The following details about a backup are available in the list: <ul style="list-style-type: none"> 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).
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, 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 from Applications list page .
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: <ul style="list-style-type: none"> 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 details about how to work with the slide out pane, see Restoring an application .

Data protection

You can protect your applications by using the IBM Storage Fusion Backup & Restore (Legacy) or Backup & Restore services.

IBM recommends the use of IBM Storage Fusion Backup & Restore service.

- [Achieving data consistency](#)

IBM Storage Fusion backs up application data consistently across all the Persistent Volumes (PVs). It ensures that the recovered data is error-free, reliable, and usable.

- [Backup and Restore](#)

Using the IBM Storage Fusion Backup & Restore service, you can backup and restore applications and workloads.

- [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

IBM Storage 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 Storage 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 Storage 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.

Backup and Restore

Using the IBM Storage Fusion Backup & Restore service, you can backup and restore applications and workloads.

Application resources

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). From 2.8 release, the Block mode volumes (**volumeMode: Block**) support is available as a technical preview.

Recipes for VMs are not supported.

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.

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.

Ensure that S3 is not in the same cluster or namespace as IBM Storage 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.

Backup policies

You can define how often backups are taken, where they get stored, and how long they are retained.

For more information about how to restore applications, see [Overview](#) and [Restoring an application](#).

For more information about OpenShift Container Platform applications, see <https://docs.openshift.com/>.

- [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.

- [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.

- [Prerequisites](#)

Ensure that you meet the prerequisites before you work with Backup & Restore service.

- [Backup & restore configuration parameters](#)

Configuration parameters in ConfigMap guardian-configmap can be used to change defaults for IBM Storage Fusion Backup & restore agent:

- [Backup & Restore Block PVC Technical preview](#)

In IBM Storage Fusion 2.8 release, support is included for block volume mode PVs as a technical preview. Before 2.8, the Backup & Restore service only supported the backup of applications with filesystem volume mode PVs.

- [Reconnecting OpenShift Container Platform cluster](#)

The OpenShift Container Platform cluster that is connected in a disaster recovery set up 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.

- [Overview](#)

The Overview page provides quick links to the important Backup & Restore service tasks to protect your applications from data corruption and loss.

- [Connecting and managing cluster connections](#)
The Topology page displays the clusters' connections.
- [Protecting your applications](#)
To protect your applications, create storage location, storage policy, and then enroll your applications for backup and restore.
- [Service protection](#)
The IBM Storage Fusion Backup & Restore service protection involves the backup of control plane to a S3 bucket. In the event of cluster failure, you can use this feature to restore the Backup & Restore service to another cluster. Configure service protection and run the initial service backup.
- [Backup and restore commands](#)
For application workloads, use these `oc` commands to create and manage backup storage location, backup policy, restore backup, and application.
- [Monitoring and managing Backup & restore service from OpenShift Container Platform](#)
You can monitor and manage Backup & restore service related operations from the OpenShift Container Platform console.

Hub and spoke model overview

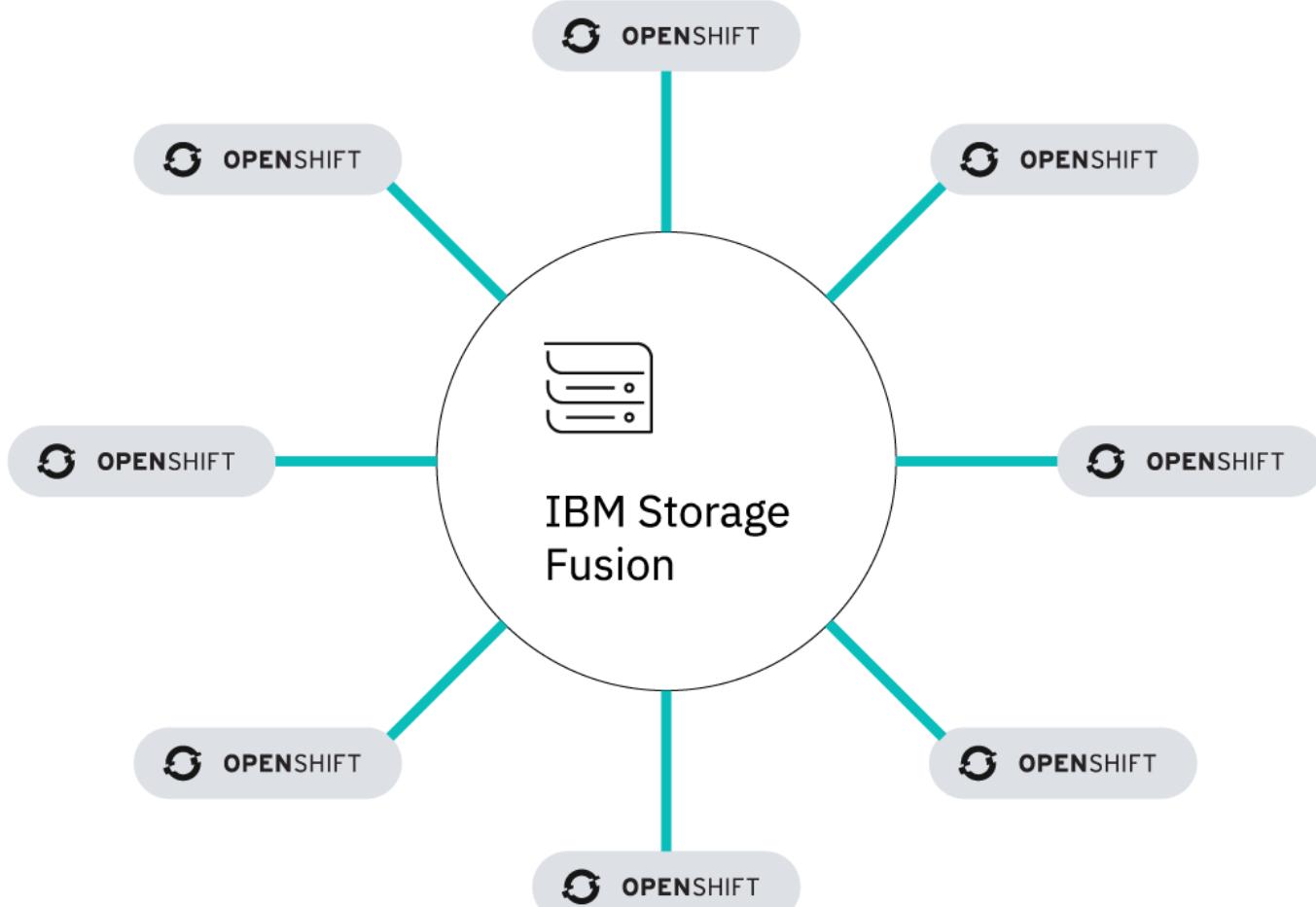
Hub and spoke model allows a single administrator to manage backups for multiple applications that exist on the different clusters.

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 Storage Fusion and the Backup & restore Agent Service on that cluster. As part of the spoke agent installation process, a communication gets 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 Storage Fusion takes care of the needed steps to auto-renew a certificate before it expires.

IBM Storage 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 Storage 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 backed 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.

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 Storage Fusion Backup & Restore service is FIPS compliant. It uses FIPS validated cryptographic modules provided by Red Hat Enterprise Linux OS/CoreOS (RHCOS).

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 [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 StorageClass:

```
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 to run successfully, ensure that in all platforms, the Backup & Restore (Legacy) and Backup & Restore services are only supported with CSI compliant storage classes that has `volumeBindingMode: Immediate`.
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 configuration parameters

Configuration parameters in ConfigMap `guardian-configmap` can be used to change defaults for IBM Storage 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 20 minutes, 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 20 minutes, 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`

The number of processing threads in the transaction manager. 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 Storage 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.

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 Storage 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 4.12 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 & Restore Block PVC Technical preview

In IBM Storage Fusion 2.8 release, support is included for block volume mode PVs as a technical preview. Before 2.8, the Backup & Restore service only supported the backup of applications with filesystem volume mode PVs.

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.

The detection and processing of Block volume mode PVs is automatic with no user configuration or changes to the input.

The CSI does not support the detection of changes in the Block volume mode PVs. As a result, every backup must read the entire Block volume mode PV to determine the incremental changes from the prior backups. The amount of time depends on the PV size. The resulting backup that is stored on S3 is still incremental and compressed.

Example Block volume mode PersistentVolume definition.

Note: The `volumeMode` designates the `PersistentVolume` as `Filesystem` or `Block`.

```
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: 9abbala3-6cd9-4d61-9f3e-95b29dab2094
    apiVersion: v1
    resourceVersion: '71161'
  persistentVolumeReclaimPolicy: Delete
  storageClassName: mylocalvs
  volumeMode: Block
```

Reconnecting OpenShift Container Platform cluster

The OpenShift® Container Platform cluster that is connected in a disaster recovery set up 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. Clean the connection.

For the procedure to clean the connection, see [Disabling the connection](#).

2. Setup the connection between clusters.

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:

```
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>
```

Create a 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>
```

Overview

The Overview page provides quick links to the important Backup & Restore service tasks to protect your applications from data corruption and loss.

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 [Data protection](#).

Application mobility

Move applications between different clusters by utilizing 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 [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](#).

Connecting and managing cluster connections

The Topology page displays the clusters' connections.

From the Topology you can do the following actions:

- [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	<p>a. On your hub cluster, go to the Topology.</p> <p>b. Click Connect clusters.</p> <p>c. Select Use the Fusion UI. The connect your cluster > Use the Fusion UI window appears. The YAML starts to generate.</p> <p>d. After the YAML is generated successfully, click Copy snippet.</p> <p>e. Go to the Spoke cluster and install IBM Storage Fusion.</p> <p>f. Go to the Services page and install Backup & Restore Agent.</p> <p>g. Paste the copied snippet to proceed with the agent install.</p> <p>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.</p>
Automate deployment	<p>To connect multiple Spoke clusters:</p> <p>a. On your hub cluster, go to the Topology.</p> <p>b. Click Connect clusters.</p> <p>c. Select Automate deployment. The connect your cluster > Automate deployment window appears.</p> <p>d. Select the storage class to use on the spoke where you want to run the YAML.</p> <p>e. Click Generate.</p> <p>The YAML starts to generate. Here, you can either use the Deploy using YAML or Deploy using oc command option.</p> <p>f. In the Deploy using YAML option, copy or download the YAML.</p> <p>g. Apply the YAML on each of the Spoke clusters.</p>

Actions on a cluster record

The Topology page lists the hub cluster and all spoke connections.

Element or section	Description or action
Cluster list	<p>The Clusters table has the following columns:</p> <ul style="list-style-type: none">• Name The name of the cluster.• Type Is it a hub or spoke cluster?• Connection status The connection status to the remote cluster as seen by the current cluster. It has connected or disconnected values.• Service status The Backup & Restore Service status. It includes Normal, Critical, Unknown, Degraded, Update available, and Updating values.• Version The version of the remote cluster's Backup & Restore service.• Backed up apps Ratio of the number of applications backed up with policies on the cluster versus the total number of applications on the cluster.• Success rate Success rate is the backup success rate. Anything less than 98% is a warning and anything less than 95% is a failure.
Search records	See Search for connections .
Actions on a connection record	<p>Click the ellipsis overflow menu of a record and select any of the following actions:</p> <ul style="list-style-type: none">• 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.• 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.• Remove connection See Remove connections

Search for connections

You can search the records by using any of the following methods:

- Sort the records by using the Type column
- Filter records based on Connection or Service.
- Enter keywords in the Search text to search the table.

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.

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
```

Protecting your applications

To protect your applications, create storage location, storage policy, and then enroll your applications for backup and restore.

Note: The label `velero.io/exclude-from-backup=true` is not supported by the Backup & Restore service.

- [Backing up applications](#)
You create backup storage locations and policies as a prerequisite to backup. Assign the policies to applications such that backups run based on the defined policy schedule.
- [Jobs](#)
The Backup & Restore > Jobs page shows all the details that are related to backup or restore jobs. It shows the history of previous and currently running jobs.
- [Managing applications backups](#)
From the Backed up applications page, you can monitor the Backup status, enroll new applications for backup, view application details, manage policies, initiate backup jobs, and restore backed up applications.
- [Restoring an application](#)
Restore a backed up application.

Backing up applications

You create backup storage locations and policies 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.

- [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.
- [Selecting applications for protection](#)
You can select applications that you want to protect. When you set up your backup hub and spoke initially, the application list is empty.
- [Backup now](#)
The Backup now initiates an immediate backup of one or more applications.
- [Managing backup policy](#)
Click Manage policies to manage policies for an application and initiate backup now.

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 Storage Fusion uses a secure connection to transfer the backups to the object storage provider of your choice.

To manage the backup location by using CRs, see [Backup and restore commands](#).

From the Locations page, you can configure at least one location to store your backed-up data. It supports a variety of object storage types.

The Locations page includes the following details:

Table 1. Location page details

Element	Description
Locations table	<p>It lists the following location details:</p> <ul style="list-style-type: none">• The Name of the location.• The Status of the location.• The Type of the location.• The Used capacity of the location.• The Policies associated to the application.• The Applications associated with the location. <p>Click the Name link to view the details of the location in a slide out pane.</p>
Customize the locations table view	<p>A list of all locations is displayed in either tile or table mode. Click the appropriate icon to toggle between the two different views.</p> <p>Note: The tile icon has four small boxes and the table icon shows a bullet list.</p> <ul style="list-style-type: none">• Tile mode<ul style="list-style-type: none">It lists all the configured locations as clickable tiles. As an overview, individual tiles provide the status of the location. For example, Connected with a green checkmark indicates that the location is online. It also displays the used capacity value, number of policies, and number of applications associated to it. Click the tiles to view the details of the location in a slide out pane.• Table mode<ul style="list-style-type: none">It lists all locations in a table format. <p>You can choose a number in the Items per page to decide the rendering of policy records per page. Use the arrows to move between pages. You can also select a page number to directly jump to it.</p>
Search the locations table	<p>Use the Search text box to filter the records in the list. You can also use the settings icon to choose the columns to display. Click Reset to default if you want the original system settings.</p> <p>In addition, you can also filter the records based on Status and Type. Choose the option All from the Status and Type drop down lists to view all records.</p>
Actions from the ellipsis overflow menu of a location record	<ul style="list-style-type: none">• You can do the following operations from the ellipsis overflow menu of the location record both in tile and table mode:<ul style="list-style-type: none">◦ Click Details to open the <Location name> slide out pane. Alternatively, you can click the Name link to view the details of the location in a slide out pane. <p>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 Policies and Applications.</p> <p>Note: The Region is only applicable for AWS locations.</p> <p>Expand the Policies and Applications to see a list of associated policies and applications. The number that is mentioned within parenthesis for Policies and Applications indicate the total number of associated policies and applications respectively.</p> <p>Click the edit icon to update Access key and Secret key. For Azure, the values are Account name and Account key.</p> <p>Click the launch YAML icon to view the YAML.</p> <ul style="list-style-type: none">◦ 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. Note: You cannot update bucket details except credentials because two backup storage locations with different names can share the same credentials and endpoint.◦ 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 Storage Fusion. However, you can still access this backup location through your storage service provider.
Add Location	Click Add Location to add a new location. For more information about the procedure, see Adding a backup storage location .

- [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 Storage Fusion user interface.

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 Storage Fusion cluster.

```
openssl s_client -connect minio-minio-ns.apps.myfusionhostname.mydomain:443 -showcerts \
| sed -n '/BEGIN CERTIFICATE/,/END CERTIFICATE/p' > tls.crt
```

2. Run the `oc` command to create a generic secret in the IBM Storage 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 Storage 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 Storage 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.

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 Storage 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 IBM Cloud (IBM Object Storage), Azure (Microsoft Object Storage), AWS (Amazon Object Storage), Spectrum Protect, MCG or Noobaa, and S3Compliant (Any Object Storage).
For the procedure to add MCG or Noobaa location, see [Using MCG or Noobaa as a backup storage location](#). For the procedure to add Spectrum 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 Location type section, enter the following credentials to connect IBM Storage 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 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.

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 Storage 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 Storage 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 Storage 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.

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 Storage 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 Storage Fusion.

Backup policies

From the Policies page, you can create new policies and manage existing policies.

Prerequisites

You must first configure the target backup location before you configure a policy.

To know more about configuring the target backup storage Location, see [Backup storage locations](#).

Table 1.

Policies table	<ul style="list-style-type: none">• The tab lists all available policies along with these details:<ul style="list-style-type: none">◦ The Name of the policy.◦ The Backup location where the backed up data would reside.◦ The Frequency of backup. It can be weekly, daily, hourly, or monthly.◦ The Time of the backup.◦ The Retention period of the backed up data.◦ The Applications that are associated with the policy.
Configure the Policies table	<p>You can also use the settings icon to configure the table display:</p> <ul style="list-style-type: none">• Choose the columns to display.• Select a row height in pixels.• Click Reset to default if you want the original system settings. <p>You can choose a number in the Items per page to decide the rendering of policy records per page. You can also select a page number to directly jump to it.</p>
Search the Policies table	Search the policy records based on the backup location. You can also enter the keywords and filter the records.

Actions from the ellipsis overflow menu of a policy record	<ul style="list-style-type: none"> Details Click Details from the ellipsis overflow menu of a policy record. The policy details slide out pane is displayed. Alternatively, you can click the policy name record to open the details. From the details slide out pane, you can do the following actions: <ul style="list-style-type: none"> View the Schedule and Location details. It also includes the associated applications. Click the Location name link to view its details. Click the YAML icon to generate it. Edit details: Click the edit icon in the slide out pane. The Edit <policy name> window is displayed. 1. From the Schedule tab, edit Frequency, Time, and Retention. 2. From the Location tab, select whether the location is In place snapshot or Object storage. If you select Object storage, then search for the location and select it. 3. Click Save. Click the delete icon in the slide out pane to delete the policy record. Edit The Edit option in the ellipsis overflow menu of a policy record is same operation as the edit icon in the details slide out pane. Delete The Delete option in the ellipsis overflow menu of a policy record is same operation as the delete icon in the details slide out pane.
Add policy	For the procedure to add a backup policy, see Creating backup policy .
Backup commands	To manage the backup polices and backup location by using CRs, see Backup and restore commands .

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.

Procedure

- Log in to IBM Storage Fusion user interface.
- From the menu, click Backup & Restore > Policies.
- Click Add policy.
- The Create a backup policy slide out pane displays on the screen.
- In the Create a backup policy pane, enter the following details to create a new backup policy that meets your business needs:
 - Enter the Policy name.
 - Enter the Frequency details:
 - 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.
 - 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 choose from a list of available object storage. You can also use the Search option to filter the location records.
 - 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.

Setting up backup policies from Applications list page

You can set up backup 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.

Before you begin

For backup and restore to work, you must create volume snapshot class.

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.

Selecting applications for protection

You can select applications that you want to protect. When you set up your backup hub and spoke initially, the application list is empty.

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 for 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.

Backup now

The Backup now initiates an immediate backup of one or more applications.

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 easily 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

- Go to Backup & Restore > Backedup applications page.
- Select one or more applications from the list.

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 <code>Initiating backup Starting backup on [app name]</code> 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 <code>Initiating backup on [n] applications</code> notification is displayed. In the Backedup applications page, you can observe that the Backup status of the selected applications are in progress.

Managing backup policy

Click Manage policies to manage policies for an application and initiate backup now.

Procedure

- Go to Backup & Restore > Backedup applications page.
- 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: <code>Mixed policies the application selected do not share the same set of policies.</code>

- To remove all policies, do the following steps:
 - In the Manage policies page, click the Remove all policies toggle button.
 - If you set the Remove all policies toggle button to on, then the policies table is disabled.
 - In the Remove all assigned policies confirmation window, click OK to proceed.
- Click Backup now to initiate an immediate backup of one or more applications.
- Select one or more policies and click Save.

Jobs

The Backup & Restore > Jobs page shows all the details that are related to backup or restore jobs. It shows the history of previous and currently running jobs.

The Jobs tab includes following details:

Table 1. Jobs page section details

Element	Description

Element	Description
Summary section	<ul style="list-style-type: none"> Success rate is the percentage of completed jobs to canceled jobs and failed jobs. For example, if completed jobs are 10 and failed jobs and canceled jobs are 0, the success rate is shown as 100%. You can see the number of jobs that are completed, canceled, and failed. Click the Time filter option and select a time. It shows the number of completed, canceled, and failed jobs along with the success rate for the selected time period.
Queue section	You can see the number of jobs in progress and pending states.
Backups tab	<p>You can see the statuses of all jobs (failed, pending, in-progress, canceled, completed). The Backups table lists all the jobs along with the following details:</p> <ul style="list-style-type: none"> The Name of the respective job. Click the Name to view job details. It takes you to the job details page. For more information, see Jobs details. Note: The batch option is unavailable. The Cluster to which the job belongs. The Application that is associated with the job. The Status of the jobs. The table shows all job statuses (failed, pending, in-progress, canceled, completed). The Started column shows the date and time of job creation. The Elapsed time of the jobs. This time is only shown for jobs in progress. The Finished column shows the date and time of job completion. The Policy column shows the policy name of the jobs.
Search Backups	<ul style="list-style-type: none"> Click the  (filter icon) to view the specified jobs according to your preferences. You can filter and view the required jobs in the jobs table. <ul style="list-style-type: none"> Choose the job Status, Policy, and Application, Cluster, and Finished time of the job. Click Apply. This lists the jobs according to your preferences. In the Backups tab table, you can sort the jobs based on Started field.
Restores tab	<p>You can see the all job statuses (failed, pending, in-progress, canceled, completed). The job Restores table lists all restore jobs along with the following details:</p> <ul style="list-style-type: none"> The Name of the respective job. Click the Name to view job details. It takes you to the job details page. For more information, see Jobs details. The Cluster to which the job belongs. The Application associated with the job. The Status of the jobs. The table shows only completed, failed, and canceled jobs. The Started column shows the start date and time of a job. The Elapsed time of the jobs. The time is displayed as Hours:Minutes:seconds. The Finished column shows the completed date and time of a job. The Policy column shows the policy name to the jobs.
Search Restores	<ul style="list-style-type: none"> Click the  (filter icon) to view the history of specified jobs according to your preferences. You can filter and view the history of required jobs in the history table. <ul style="list-style-type: none"> Choose the job Status, Policy, Application, Cluster, and Finished time of the job. Click Apply. This lists the jobs according to your preferences. In the Restore tab table, you can sort the jobs based on the Finished status.
Cancel jobs	<p>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.</p> <pre>spec: jobControl: cancel</pre> <p>If you cancel during backup, the volume snapshots get removed as a part of cleanup.</p> <p>If you issue a cancel during restore, the cleanup happens only when you choose a new or an existing namespace.</p> <p>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.</p>

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 scale out from the OpenShift® Container Platform console, do the following steps:

- Change the project to the Backup & Restore install namespace. The default is `ibm-backup-restore`.
 - Search and select the `transaction-manager` deployment.
 - 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.
- [Jobs details](#)**
- You can view the complete status or stages of backup or restore jobs from the IBM Storage Fusion user interface. It includes job details such as job status, workloads, storage capacity, resources, backup sequences, and backup data transfer related information.

Jobs details

You can view the complete status or stages of backup or restore jobs from the IBM Storage Fusion user interface. It includes job details such as job status, workloads, storage capacity, resources, backup sequences, and backup data transfer related information.

Table 1. Job details

Element	Description
Details section	In the Details section, you can view the following details. <ul style="list-style-type: none"> The Duration of the jobs. The Created field shows the time that is taken to complete the job. The Size shows the actual size of the backup or restore jobs. The Policy shows the policy that is assigned to the job. The Application shows the linked application for the job. The Backup ID shows the cluster ID of the respective job. The Backup Location shows the actual location of the job.
Summary section	In the Summary section, the stages of the job are shown in accordion mode. Note: By default the accordion mode is selected and the step currently in progress is expanded. After the job is completed, you can expand all the three stages and see the details. You can also toggle through the clipboard icon to view the job traces.
	<p>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.</p> <p>Backup sequence You can see the status of pre-hook, backing up PVCs, post-hook, and backing up etcd resources.</p> <p>Data transfer You can view backup location, backup size, upload speed, and the status of upload. Note: You can view the data transfer details only for the jobs such as backups to S3 object and restore from S3 object.</p>
Job stages	Backup jobs follow the Inventory, Backup sequence, and Data transfer order. Restore jobs follows the Data transfer, Restore sequence, and Inventory order.
Actions	<ul style="list-style-type: none"> Click Collect logs to collect logs. Click Delete job to delete failed jobs. Use the <code>DeleteBackupRequest</code> CR to manually delete successful backups. Click Launch YAML to view the CR's YAML in the OpenShift® console.

Managing applications backups

From the Backed up applications page, you can monitor the Backup status, enroll new applications for backup, view application details, manage policies, initiate backup jobs, and restore backed up applications.

Section	Description
Backup status	From the Backup status section, you can select the range of clusters that you want to monitor. The output is represented pictorially in a circular chart with much percentage of backup is completed, in progress, canceled, and failed.
Application list	The Applications page lists all your OpenShift® applications with their backup details, such as Name, Cluster, Backup status, Last backup on, Success rate
Protect applications	When you set up your backup hub and spoke initially, the application list is empty. For the procedure to populate the application list or add protection to a protection .
Search records	You can search the records by using any of the following methods: <ul style="list-style-type: none"> Filter records based on Cluster or Status. Enter key words in the Search text to search the table.

Section	Description
Actions on an application record	<p>You can select one or more applications at a time. The following options are available in the ellipsis overflow menu:</p> <ul style="list-style-type: none"> View details If the application is in the local cluster, then the application details page is displayed. For more information about the application details page, see View application details. If the application is in a remote spoke cluster, then the application details include the following details: <ul style="list-style-type: none"> Usage section having Available backups and used capacity in GiB. Backup policies section where you can view the assigned policies. Backup policies section of an application from an Agent cluster allows to List of backups available for the application. It includes Time, Status, Policy, Location, and Size (GiB) columns. From the ellipsis overflow menu of individual backups, you can View details, Restore, and Delete. <p>The following batch operations can be done on the backup records:</p> <ul style="list-style-type: none"> Restore Click Restore to restore all the backups of the selected remote cluster application. <p>Completed and unexpired backups of applications with an object storage location can be restored. In place snapshots cannot be restored if the connection was disconnected. However, backups with object storage location like S3 can be restored to another cluster.</p> <p>If you want to restore object storage or S3 backup after the removal of the connection, then the following notification appears in the footer:</p> <p>Original cluster removed The original cluster where this app was deployed, has been removed. To restore this application, choose a new cluster from the Actions submenu.</p> <ul style="list-style-type: none"> Backup now From the Actions submenu, click Backup now to initiate an immediate backup of the application. The option is disabled for an application if the connection is removed. Manage policies Click Manage policies to manage policies for an application and initiate backup now. For more details about the procedure, see Manage application policies. <p>This option is available only when you select a single application.</p> <ul style="list-style-type: none"> Backup now Click Backup now to initiate an immediate backup of one or more applications. For more details about the procedure, see Backup now. Restore Restores an application. This option is available only when you select a single application. <p>After you remove a Agent or Spoke connection from the Hub cluster, it does not show up in the Topology page. Also, you can restore the available (available) applications from the Hub cluster to a different cluster.</p> <p>If you want to restore object storage or S3 backup after the removal of the connection, then the following notification appears in the Restore application:</p> <p>Cluster removed This application no longer has a connected cluster. To restore this application, choose a new cluster from the Actions submenu.</p> <ul style="list-style-type: none"> Delete For failed records, choose the Delete option. For other backups, manually create and issue a <code>DeleteBackupRequest</code> CR.

Restoring an application

Restore a backed up application.

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 can restore backup data to a different `StorageClass` provisioner on an alternative 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 or command prompt.
 - 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 Cloud Pak for Data.
- You must have OpenShift® APIs for Data Protection (OADP) version 1.3.1.

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 Cloud Pak for Data.
- 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`.

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>, Restore destination cluster where the application must be restored.
 - Same cluster
 - New cluster

Note: Only S3 backups can be restored to a different cluster. If you do not have Spoke clusters, this option is not available.

The details in the Summary section vary based on your selections.

5. Select the destination:

Option	Description
Same cluster	In the Project destination section, you can choose to restore to any of the following locations: Use the same project the application is already using Existing project Select the project from the drop-down list. Create a 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 specify only a single namespace. You cannot restore multiple namespaces to a single alternative namespace.
New cluster	a. In the Restore destination section, select the cluster where you want to restore your application. Note: From the IBM Storage Fusion user interface, you cannot restore to a cluster that does not have an identical storage class name as the originator cluster. The same storage class name is needed in the destination or the target cluster. However, only the storage class name must be identical to the backup, while the underlying provisioner can be native to the target cluster. If the identical storage class is not found, the default storage class is used. If you want to restore backup data to a different StorageClass provider, do this through the command line or command prompt. See <i>Before you begin</i> section of this topic to know how to use the <code>targetStorageClass</code> field in the Restore CR.

6. Click Next.
7. If you have not selected the backups, then select the backups that you want to restore and click Next.
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. 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 changes to Completed.
11. 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.

Service protection

The IBM Storage Fusion Backup & Restore service protection involves the backup of control plane to a S3 bucket. In the event of cluster failure, you can use this feature to restore the Backup & Restore service to another cluster. Configure service protection and run the initial service backup.

- [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

- IBM Storage Fusion Backup & Restore (Legacy) service is not supported after a service recovery as it replaces all existing objects with the objects in the service backup.
- In Backup & Restore, one hub cluster and more than one spoke clusters can be connected to easily 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 Storage 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 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 initiate 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 Storage 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 Storage Fusion namespace must remain the same between the service backup and service restore. When you recover a service backup to a new IBM Storage 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.

Option	Description
No service backups available	If no service backups are available, then No backups discovered message is displayed. 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: <ul style="list-style-type: none">• 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 icon ().

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 (Legacy)	No action required
Empty cluster but Backup & Restore (Legacy) 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 (Legacy). However, any available Backup & Restore (Legacy) backups 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

- 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.
- 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.
- 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.
- 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.

Backup and restore commands

For application workloads, use these `oc` commands to create and manage backup storage location, backup policy, restore backup, and application.

If you are not currently in the same namespace, then include `-n <namespace>` in all commands. If you are already in the IBM Storage Fusion namespace (`ibm-spectrum-fusion-ns`), then `-n <namespace>` is not required.

Backup storage location

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: If backup policies exist that use the backup storage location, then you cannot delete it.

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.bnrt-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
```

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
```

Monitoring and managing 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 Storage Fusion user interface.

Prerequisites

- Enable Backup & restore service in IBM Storage Fusion.
For the steps to enable Backup & restore service from IBM Storage Fusion user interface, see [Backup & Restore](#).
- After IBM Storage 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 4.14 and 4.15 versions. 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	<p>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.</p> <p>The Applications subsection includes the following details:</p> <ul style="list-style-type: none"> 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	<p>The Backup & Restore jobs summary includes the following details:</p> <ul style="list-style-type: none"> 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	<p>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:</p> <ul style="list-style-type: none"> Backed up applications Applications with assigned policies Not backed up applications Applications without assigned policies <p>The Most recent backup includes the following details:</p> <ul style="list-style-type: none"> 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.

Section	Description
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: <ul style="list-style-type: none"> Edit backup policy Edit YAML Delete backup policy
Create backup policy	1. Click Create backup policy. The Create BackupPolicy page is displayed. 2. Enter the following details of the policy: <ul style="list-style-type: none"> 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 Storage 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: <ul style="list-style-type: none"> Edit credentials <ul style="list-style-type: none"> Update Endpoint, Access key, or Secret key values. 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 . <p>Note: IBM does not support the creation of two backup storage locations that have both identical endpoint and bucket names.</p> <p>1. Click Create backup location. 2. Enter the details about the backup location: <ul style="list-style-type: none"> 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 Storage 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. </p> <ul style="list-style-type: none"> 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 Storage 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	<p>From the ellipsis overflow menu of an application record, you can do the following operations:</p> <ul style="list-style-type: none"> Assign backup policy Restore <p>It opens a Restore window.</p> <ol style="list-style-type: none"> In the Select a project destination section, select from the following options: <ul style="list-style-type: none"> 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. Select a backup from the available list of backups. Optionally, select Include missing etc resource to restore any etc resources that are not present in the existing OpenShift project. 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. Click Restore.
Assign backup policy	<p>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.</p> <p>Note: IBM does not support the creation of two backup storage locations that have both identical endpoint and bucket names.</p> <ol style="list-style-type: none"> 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. Select one or more policies. If you want to initiate a backup right away, set the Run backup now toggle button to on. Click Assign. <p>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.</p>

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	<p>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.</p> <p>You can select a field and enter keywords in the Search to filter records.</p> <p>You can also choose to view the list of jobs in last 24 hours, last 7 Days, or last 30 days.</p>
Actions to perform on a jobs record	<p>The ellipsis overflow menu of a job record are as follows:</p> <ul style="list-style-type: none"> 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 Storage Fusion user interface. In the Jobs page, click Collect logs. Job summary Click Jobs summary to go to the Jobs page of the IBM Storage 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).

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.

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 Cloud Pak for Data to the same or different cluster](#)**

IBM Cloud Pak for Data is now integrated with IBM Storage Fusion , which enables you to create online backups and to restore them to the same cluster or to a different cluster.

- [IBM Storage Fusion repository for recipes](#)

The IBM Storage Fusion public GitHub repository provides helpful utilities for IBM Storage Fusion and allows members of the broader IBM Storage 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.

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: demoechohook/pre
- group: mysql_data
- hook: demoechohook/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:

- **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 volume 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**.

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.
3. **check**: check for conditions on workload resources (such as **readiness**, **replicaCount** and so on).

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). 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`: if specified, then the workload resource needs to match this label selector.
 - 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, the default is 300 seconds.

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.

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:

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/v1alpha1
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
```

Backup and restore of IBM Cloud Pak for Data to the same or different cluster

IBM Cloud Pak for Data is now integrated with IBM Storage Fusion , which enables you to create online backups and to restore them to the same cluster or to a different cluster.

Note: IBM Cloud Pak for Data supports backup and restore on both x86 and Power.

To create IBM Cloud Pak for Data backups with IBM Storage Fusion , see [Creating and scheduling online backups of Cloud Pak for Data with IBM Storage Fusion](#).

To restore IBM Cloud Pak for Data online backup and restore to the same cluster with IBM Storage Fusion , see [Cloud Pak for Data online backup and restore to the same cluster with IBM Storage Fusion](#).

To restore IBM Cloud Pak for Data backups to a different cluster with IBM Storage Fusion , see [Cloud Pak for Data online backup and restore to a different cluster with IBM Storage 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 Cloud Pak for Data.

IBM Storage Fusion repository for recipes

The IBM Storage Fusion public GitHub repository provides helpful utilities for IBM Storage Fusion and allows members of the broader IBM Storage Fusion community to share and contribute tools and knowledge.

This repository hosts the following categories of artifacts that can be used with IBM Storage Fusion:

- IBM Storage Fusion Backup and Restore recipes
- Helpful utilities and scripts

For example, you can run scripts from the command line to view recipe workflow logs as well as a complete list of resources backed up or restored.

For the IBM Storage Fusion GitHub repository, see [storage-fusion GitHub repository](#).

The IBM Storage Fusion development team has provided a number of helpful recipes to orchestrate backup and recovery of popular applications with the IBM Storage Fusion Backup & restore service. The repository includes recipes for applications such as IBM Db2, PostgreSQL, Postgres EnterpriseDB, MongoDB, Redis, and others. Check the repository for the complete inventory of applications.

If you like to develop recipes for other popular applications, read the following blog series on creating backup and restore recipes:

1. [How IBM Storage Fusion Simplifies Backup and Recovery of Complex OpenShift Applications - Part 1 "So Why is this Complex?"](#)
2. [How IBM Storage Fusion Simplifies Backup and Recovery of Complex OpenShift Applications - Part 2 "Orchestrations and Recipes"](#)
3. [How IBM Storage 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](#).

Data Cataloging

Companies need the ability to use unstructured data to meet their business priorities.

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 easily connects to these data sources to rapidly ingest, consolidate, and index metadata for billions of files and objects.

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.

Note: The IBM Spectrum® Discover service name is used interchangeably with Data Cataloging.

Benefits of Data Cataloging

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 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.

- [Data Cataloging architecture](#)

Data Cataloging is an extensible platform that provides exabyte scale data ingest, data visualization, data activation, and business-oriented data mapping.

- [Planning](#)

- [Configuring Data Cataloging](#)
This section provides information on how to deploy and configure Data Cataloging capabilities.
- [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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover to perform retention analytics on archive data managed by IBM Storage Protect.
- [Managing tags](#)
A tag is a custom metadata field that is used to supplement storage system metadata with organization-specific information. For instance, an organization might segment their storage by project or by chargeback department. Those facets do not show up in the system metadata. Additionally, the storage systems themselves do not provide management and reporting capabilities based on those organizational concepts. Use custom tags to store additional information and manage, report, or search for data by using that organizationally important information.
- [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 Spectrum Discover 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 sanitization, and more. Data is identified by IBM Spectrum Discover 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 Spectrum Discover, which might or might not include enriched metadata for the records. If it does include enriched metadata, IBM Spectrum Discover catalogs that metadata and makes it immediately searchable.
- [Using the IBM Spectrum Discover application catalog](#)
Use the IBM Spectrum Discover application catalog to search, download, or deploy applications (which are provided by IBM®, customers, or third parties) to use in IBM Spectrum Discover.
- [Reports](#)
Reports can be generated upon applying tags to a set of data.
- [High availability for a Db2 Warehouse MPP deployment](#)
For an MPP deployment, Db2® Warehouse provides high availability, offering you the ability to have your data warehouse carry on with its activities if failures occur.
- [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 Data Cataloging service environment](#)
You can monitor the health and status of the Data Cataloging service environment and obtain audit log information.
- [Updating the network configuration](#)
This topic describes how to update the network configuration.
- [Using a third-party data movement application to manage data](#)
Introduction to data movement with IBM Spectrum Discover.
- [Enabling feature for skipping the snapshot directories](#)
Enabling skipping feature allows you to skip the metadata ingestion to the snapshot directories.
- [Data protection](#)
- [REST API for Data Cataloging](#)
- [Graceful shutdown](#)
Provides detailed instructions to put Data Cataloging in idle state on an OpenShift® environment.
- [Health check monitoring](#)
Provides series of checkpoints to check Data Cataloging health status.
- [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.
- [Collecting logs and metrics](#)
Steps to collect logs and metrics for the Data Cataloging.
- [Creating a Data Cataloging application for metadata-based policies](#)
Provides information about how to create Data Cataloging application for metadata-based policies.
- [Data Cataloging Harvester CLI](#)
Data Cataloging Harvester is a new capability that is designed to import external data to the 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 Data Cataloging.

Data Cataloging architecture

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

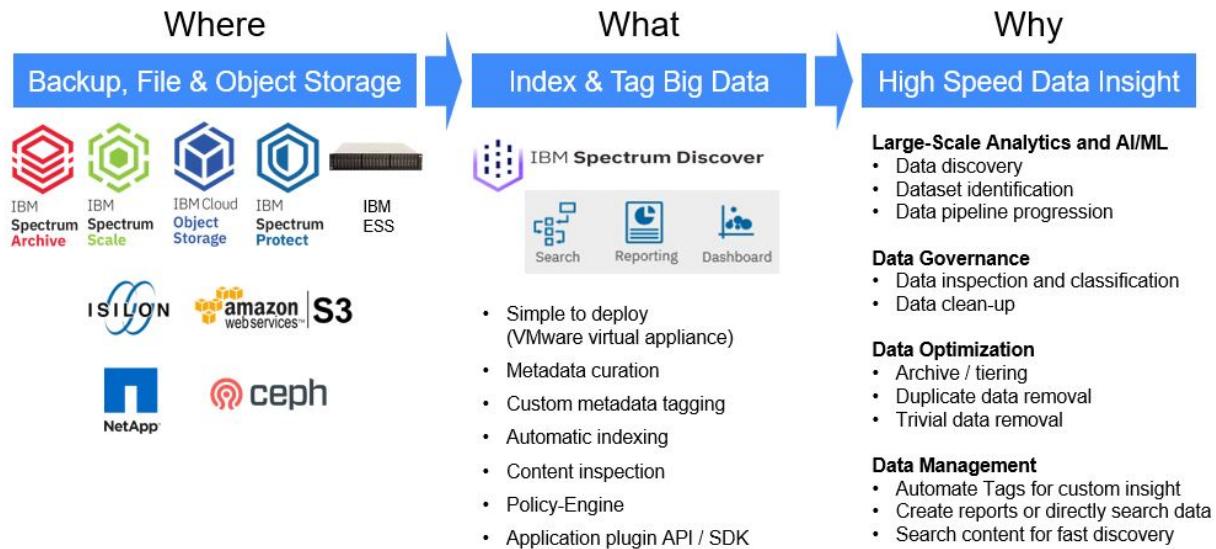
Business-oriented data mapping

- System-level data tagging
- Contextual data tagging
- Policy-driven workflows

The following figure illustrates a high-level view of the Data Cataloging architecture.

Figure 1. Data Cataloging architecture

IBM Spectrum Discover Overview



Data management

Data Cataloging connects to the data sources shown in the architecture image (*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.

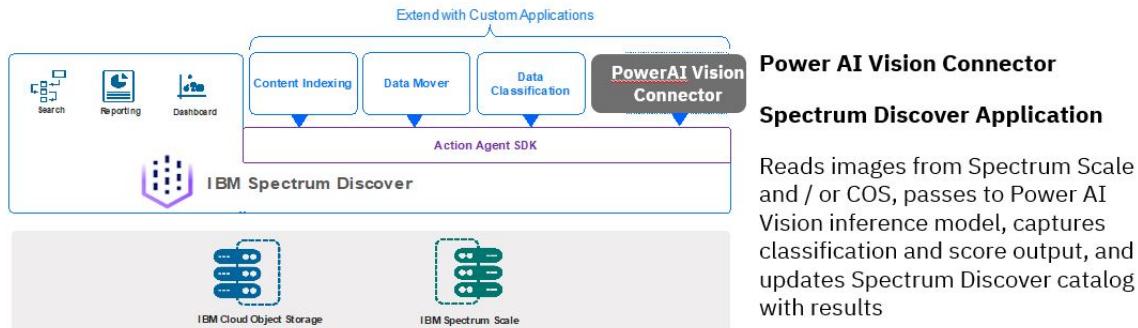
After the data is ingested, analytics are automatically applied to classify and group the data according to the different system metadata attributes. The data can be inspected automatically in Data Cataloging by using built-in content search capabilities to identify sensitive and personally identifiable 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 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 Data Cataloging catalog to gain insight about your data and to find your data easily.

The Data Cataloging architecture also supports a community-supported catalog of open source applications that enhance and customize the capabilities of 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 Data Cataloging application catalog* in the Administration section.

Figure 2. Application SDK architecture



- **[Role-based access control](#)**
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 Data Cataloging.
- **[Cataloging metadata](#)**
System metadata is created and updated by the host system, and not the application software. Data Cataloging allows the addition of tags that can capture non-system metadata-specific attributes, which are stored in the Data Cataloging catalog.
- **[Enriching metadata](#)**
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 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 Data Cataloging](#)**
Reports for 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

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 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 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 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 Spectrum® Discover, 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 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 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. Data Cataloging allows the addition of tags that can capture non-system metadata-specific attributes, which are stored in the 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.

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 Data Cataloging. 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 Data Cataloging to capture file system event notifications and deliver them to Data Cataloging by using Kafka.

Important: If the connection from IBM Storage Scale to Data Cataloging is interrupted, the watch suspends. Additionally, events are no longer captured in Data Cataloging, which requires a file system rescan to capture the lost updates.

Enriching metadata

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 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 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 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 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 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 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 Spectrum® Discover 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 Spectrum® Discover 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 Spectrum Discover to extract content from source data and index it automatically into the IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum® Discover 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 Spectrum Discover.

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 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 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 Data Cataloging, an LDAP server, or the IBM Cloud® Object Storage System. The administrator can manage the user access functions.

Roles

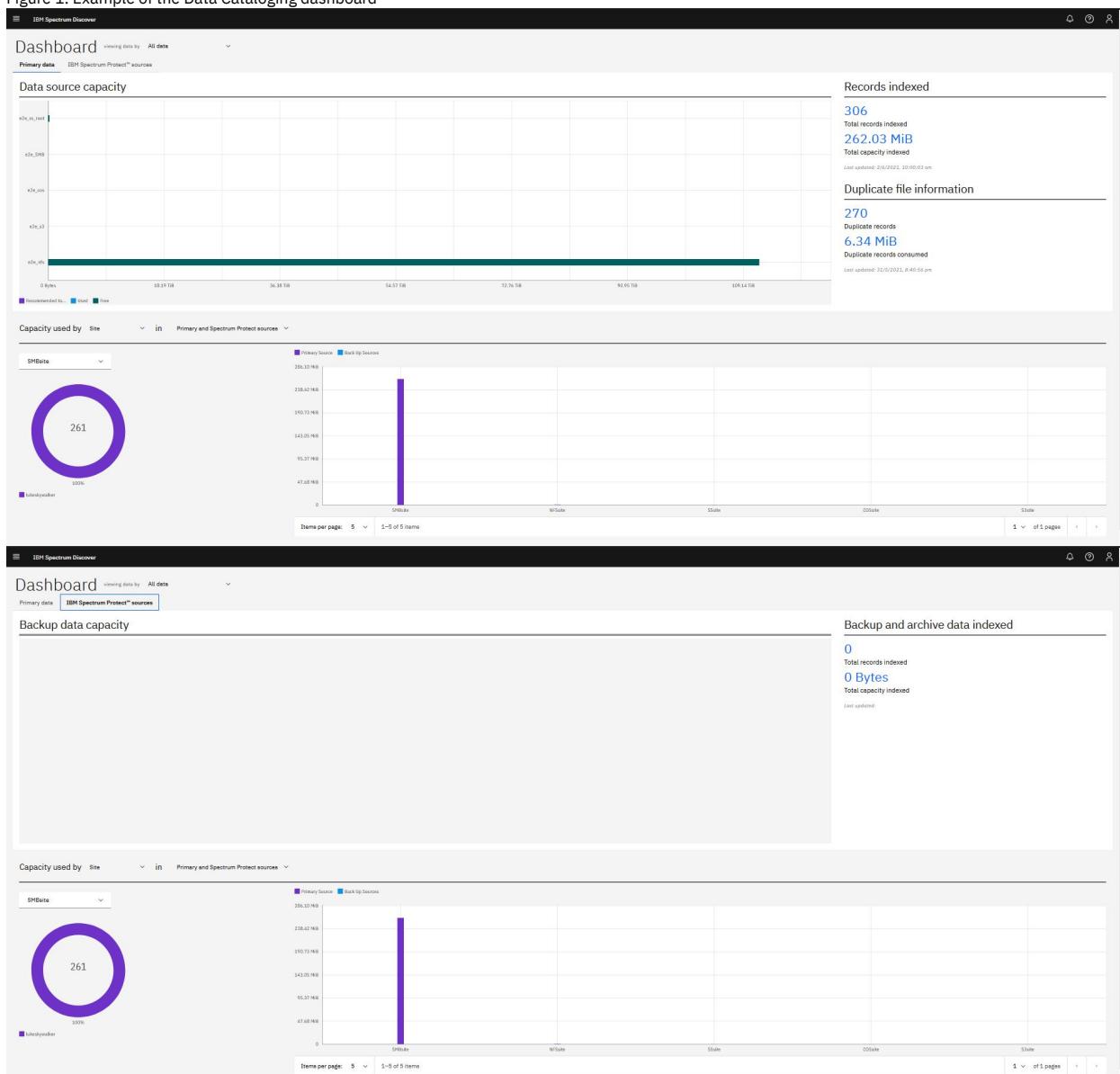
Roles determine how users and groups can access records on the Data Cataloging environment.

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 Data Cataloging dashboard is shown.

Figure 1. Example of the 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

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 Data Cataloging

Reports for 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](#) in *Data Cataloging: Administration Guide*.

For more information, see [REST API for Data Cataloging](#). For more information, see *REST API* in the *Data Cataloging: REST API Guide*.

Planning

- [Data migration path from 2.0.4 OVA environment](#)

You can use the following information to process data migration from an Open Virtual Appliance (OVA) Db2 single instance to the Red Hat® OpenShift® Db2 single instance.

Data migration path from 2.0.4 OVA environment

You can use the following information to process data migration from an Open Virtual Appliance (OVA) Db2 single instance to the Red Hat® OpenShift® Db2 single instance.

Before you begin

Before you start the data migration activity, you must meet the following prerequisites:

- To make the connection, consider following two different environments:
 - An IBM Spectrum® Discover OVA single instance environment, where database will be backed up.
 - A fresh IBM Spectrum Discover Red Hat OpenShift Container Platform (OCP) single environment where the data is planned to be restored.
- Ensure to maintain consistent connection to the host by a `tty` or by `ssh` that has a consistent and permanent solution to send an alive packet every certain time. For example, `$ ssh -o ServerAliveInterval=30 user@host.com`.
- Ensure adequate storage space for the data backup and restore at the locations where the data is planned to be stored. The number of data is directly proportional to the storage required for backup and restore. For example, 100 Million records need around 200 GB of storage to perform the backup properly. Storage limitations might result in data corruption or errors during the migration process.

Note: By performing this procedure, it stops all database interactions and the database might not be accessible during the process. You must have the environment free of usage during the process.

WHAT'S NEXT

- **Backing up data**
Use Data Cataloging Open Virtual Appliance (OVA) single instance environment to back up the database.
- **Restoring data**
Use IBM Spectrum Discover Red Hat OpenShift Container Platform (OCP) single environment to restore the data.

Backing up data

Use Data Cataloging Open Virtual Appliance (OVA) single instance environment to back up the database.

Procedure

1. Establish a connection to Data Cataloging OVA backup target instance. The data is backed up by a terminal or through `ssh`, considering the need of a consistent and permanent connection during the migration process as explained in the [prerequisites](#) section.
2. Create conf file and save it (`data_migration.conf` for instance) or make sure that the following environment variables are exported in the current environment:

```
DM_ACTION=<BACKUP|RESTORE>
DM_DATA_PATH=<absolute path as working directory where the data will be stored (ensure having enough storage)>
DM_TAR_NAME=<Tarball name which will be used to locate backup or restore tarball data>
```

The following example demonstrates how to use configuration file content:

```
DM_ACTION=BACKUP
DM_DATA_PATH=/mnt/bludata0/
DM_TAR_NAME=databackup-2022
```

The following example demonstrates how to export the environment variables:

```
$ export DM_ACTION=BACKUP
$ export DM_DATA_PATH=/mnt/bludata0/
$ export DM_TAR_NAME=databackup-2022
```

As mentioned, the configuration file must contain the preceding variables and their values, or you can export those values as environment variables on the same instance where the script resides.

3. Issue the following commands as per the needed scenario to start the data backup:

In case you set up the variables on the environment, do the following steps:

```
curl -s -O https://raw.githubusercontent.com/IBM/ibm-spectrum-discover-resources/2.0.5.2/data_migration.sh
```

In case you set up a configuration file with necessary variables (`data_migration.conf` for instance), do the following steps:

```
curl -s -O https://raw.githubusercontent.com/IBM/ibm-spectrum-discover-resources/2.0.5.2/data_migration.sh
chmod +x data_migration.sh
./data_migration.sh data_migration.conf
```

After the command is issued, you get a message and also a tarball with the backed-up data as shown here:

```
DONE: Data backed up on tarball -> databackup-2022.tar.gz on current directory /home/moadmin
```

Note: Depending on your computer power and number of records, this task might take hours to complete.

4. After the data backup is complete, transfer the tarball to the Data Cataloging Red Hat® OpenShift® instance by using `scp` or any other method, specifically to the bastion node to be prepared for restoring the data.

Restoring data

Use IBM Spectrum® Discover Red Hat® OpenShift® Container Platform (OCP) single environment to restore the data.

Procedure

1. Establish a connection to IBM Spectrum Discover Red Hat OpenShift instance where the data is to be restored by a terminal or through `ssh`, considering the need of a consistent and permanent connection during the migration process as explained in the [prerequisites](#) section.
2. Create conf file and save (`data_migration.conf` for instance) or make sure that the following environment variables are set:

```
DM_ACTION=<BACKUP|RESTORE>
DM_DATA_PATH=<absolute path as working directory where the data will be stored (ensure having enough storage)>
DM_TAR_NAME=<Tarball name which will be used to locate backup or restore tarball data>
```

The following example demonstrates how to use configuration file content:

```
DM_ACTION=RESTORE
DM_DATA_PATH=/mnt/bludata0/
DM_TAR_NAME=databackup-2022
```

The following example demonstrates how to export the environment variables:

```
$ export DM_ACTION=BACKUP
$ export DM_DATA_PATH=/mnt/bludata0/
$ export DM_TAR_NAME=databackup-2022
```

As mentioned, the configuration file must contain the preceding variables and their values, or you can export those values as environment variables on the same instance where the script resides.

3. Issue the following commands to start data restore:

In case you set the variables on the environment, do the following steps:

```
curl -s -O https://raw.githubusercontent.com/IBM/ibm-spectrum-discover-resources/2.0.5.2/data_migration.sh
```

In case you set a configuration file with necessary variables (data_migration.conf for instance), do the following steps:

```
curl -s -O https://raw.githubusercontent.com/IBM/ibm-spectrum-discover-resources/2.0.5.2/data_migration.sh  
chmod +x data_migration.sh  
. ./data_migration.sh data_migration.conf
```

After the preceding command is issued, you get a message as shown here:

```
Database restored. Data migration complete!
```

Note: Depending on your computer power and number of records, this task might take hours to complete.

4. After the restoration process is complete, your data is available on the new Red Hat OpenShift IBM Spectrum Discover instance and you can continue using your environment.

Configuring Data Cataloging

This section provides information on how to deploy and configure Data Cataloging capabilities.

- [Setting up Data Cataloging custom user and password](#)

Use this information to create a custom user and password for the Data Cataloging service.

- [Configure data source connections](#)

Data source connections describe the source data systems for which 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.

- [Using custom TLS certificate](#)

You can change the TLS certificate that is used by Data Cataloging for serving web pages and the REST API endpoints.

- [IBM Spectrum Storage software requirements](#)

Use Data Cataloging to index metadata from other applications and to orchestrate the data management.

- [Data Cataloging Installation with alternative VLAN](#)

Install and configure Data Cataloging that uses additional VLAN connected through the IBM Storage Fusion upstream links.

Setting up Data Cataloging custom user and password

Use this information to create a custom user and password for the 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 Data Cataloging service from the IBM Storage Fusion user interface.

Configure data source connections

Data source connections describe the source data systems for which Data Cataloging indexes metadata.

Creating data source connections in Data Cataloging identifies source storage systems that are to be indexed by 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. 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 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 *Data Cataloging: Concepts, Planning, and Deployment Guide*.

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 Data Cataloging when they are in separate clusters.

Remember: Data Cataloging does not support file or file path names that use characters that are not part of the UTF-8 character set.

- [**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 Spectrum Discover based on values that are derived from IBM Storage 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.
- [**Data Cataloging and S3 object storage data source connections**](#)
Use this information to understand how Data Cataloging works with S3-compliant object store.
- [**Scanning an Elastic Storage Server data source connection**](#)
Use the IBM Spectrum Discover GUI to scan an Elastic Storage Server (ESS) data connection.
- [**Creating a Network File System data source connection**](#)
You can use the IBM Spectrum Discover 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 Spectrum Discover graphical user interface.
- [**Creating an IBM Storage Protect data source connection**](#)
Creating the IBM Storage Protect data connection by using the IBM Spectrum Discover graphical user interface.
- [**Configuring data source connections offline**](#)
Multiple source data connections can be added to the IBM Spectrum Discover 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*.

Data Cataloging supports the following method of scanning IBM Storage Scale data sources:

Automated scanning

A data source connection is defined on Data Cataloging, including details of how to connect to the IBM Storage Scale system. Data Cataloging connects to the IBM Storage Scale system by using these details, scans the file system and sends the details back to 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 Spectrum Discover 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 Spectrum Discover to collect system metadata from IBM Storage Scale file system.
- [**Manual scanning of an IBM Storage Scale data source**](#)
How to configure IBM Spectrum Discover 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 Spectrum® Discover 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover connects to. To update the authentication keys, follow these steps:

1. Make sure 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum® Discover 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 `a11` 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 Spectrum® Discover supports two levels of automated scanning of IBM Spectrum Scale systems. Both levels require that IBM Spectrum Discover 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 Spectrum Discover and ingested locally); Or
- If the output is, instead, pushed to the ingest Kafka queue of IBM Spectrum Discover system directly from the IBM Storage Scale policy output.

The Kafka queue of IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum® Discover 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 Spectrum® Discover 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/ltsse/bin/eadm, /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/mmafmcctl, /usr/lpp/mmfs/bin/mmafmcosctl, /usr/lpp/mmfs/bin/mmaddcallback,
/usr/lpp/mmfs/bin/mmdelecallback
```
 - 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 Spectrum® Discover 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 Spectrum Discover 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 Spectrum® Discover 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 Spectrum Discover 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 Spectrum® Discover graphical user interface to create data connections from the source storage systems.

Procedure

1. Log in to the IBM Spectrum Discover 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 Spectrum Discover.

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 Spectrum Discover 2.0.3.1 removes the support for self-generated RSA key pair for IBM Spectrum Discover. 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 Spectrum Discover 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 Spectrum Discover 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.

- 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
```

- 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 Spectrum® Discover 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 Spectrum Discover graphical user interface, the data moves asynchronously back to the IBM Spectrum Discover.

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 Spectrum Discover 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 Spectrum Discover GUI for that connection.

Note: You cannot run scans unless you add override warnings in the configuration file.

Procedure

- Log in to IBM Spectrum Discover web interface.
 - Click  menu and go to Data connections > Connections.
 - Select the data source connection name that you want to scan. Make sure that the connection is online for your system ready to scan. (There is an indicator in the State column.)
 - 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 Spectrum Discover 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 Spectrum® Discover 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum® Discover to connect to IBM Spectrum Scale. After completing these steps, data can be ingested from an IBM Spectrum Scale data source to IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover.

Procedure

1. Perform a file system scan to collect system metadata from IBM Spectrum Scale to be ingested into IBM Spectrum Discover. 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 Spectrum Discover master node. For more information, see [Copying the output of the IBM Storage Scale file system scan to the IBM Spectrum Discover master node](#).
 3. Ingest data from the file system scan in IBM Spectrum Discover. For more information, see [Ingesting metadata from IBM Storage Scale file system scan in IBM Spectrum Discover](#).
 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 Spectrum® Discover.

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 Spectrum Discover 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
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 Spectrum Discover node to the required location on the IBM Storage Scale cluster node.

1. Log in to the IBM Spectrum Discover node through Secure Shell (SSH) with the `modadmin` username and password:

```
ssh modadmin@spectrum.discover.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 Spectrum Discover 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
# mmapapplypolicy 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 [spectrumsdiscover] 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 Spectrum Discover. The `scale_connection` value must match the value that is defined in the `Data Source` column of the Data Connections page in the IBM Spectrum Discover GUI.

The `username` must be a valid name of the IBM Spectrum Discover 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 Spectrum Discover 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`

```
[spectrumsdiscover]
# Name of the Spectrum Scale connection to scan files from
# Check using the Spectrum Discover connection manager APIs
scale_connection=fs3
# Spectrum Discover Master Node IP
master_node_ip=203.0.113.23
# Spectrum Discover user name, having 'dataadmin' role
# Use format <domain_name>/<username>
# e.g. username=Scale/scaleruser1
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 Spectrum Discover user that is configured in the `scale_scanner.conf` file with the `username` under the `spectrumsdiscover` 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 Spectrum Discover 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 Spectrum Discover master node

After you have scanned your IBM Spectrum Scale file system and have the `list.metaOcean` output file, copy it to the IBM Spectrum® Discover master node.

Procedure

As an IBM Spectrum Discover 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 Spectrum Discover

Records are inserted into IBM Spectrum® Discover 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 Spectrum® Discover.

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 Spectrum® Discover based on values that are derived from IBM Storage Archive metadata to help in searching and categorizing files.

IBM Spectrum Discover 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.

resndnt

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 Spectrum Discover displays a zero if the file is moved to tape.

Remember: The migration state information is collected and summarized in the IBM Spectrum Discover State facet. You can access this facet by using IBM Spectrum Discover visual search. For more information, see [Searching](#). For more information, see the topic [Searching](#) in the *Data Cataloging: Administration Guide*.

The IBM Spectrum Discover interface displays the search results including the metadata information.

Important: The location information that is displayed in IBM Spectrum Discover 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 Spectrum® Discover cluster. When configured, the IBM Spectrum Discover consumes the events and indexes them into the IBM Spectrum Discover database.

Restriction: IBM Spectrum Discover 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 Spectrum Discover.

- **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 Data Cataloging**

Ingesting IBM Cloud Object Storage event records into 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 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 Spectrum® Discover 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

e2e_SMB selected							
Connection name	Connection type	Cluster	Data source	Site	State	Scan status	Next scan
e2e_SMB	SMB/CIFS	9.11.201.170	lukeskywalker	SMBsite	● Online	●	
e2e_cos	IBM COS	e09cdac0-80f8-73be-00ed-cb8edeede242	e2e_ui_cos	COSsite	● Online	●	
e2e_nfs	NFS	9.11.201.172	Isilon	NFSsite	● Online	●	
e2e_s3	S3	s3.eu-west-1.amazonaws.com	e2e-ui-ta	S3site	● Online	●	
e2e_ss_root	Spectrum Scale	modevmm19.tuc.stqlabs.ibm.com	scale0	SSsite	● Online	●	

Items per page: 20 | 1–5 of 5 items | 1 of 1 pages | < >

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

Add data source connection

Connection name:

Connection type:

Select a collection

Schedule data scans

Cancel | Submit Connection

4. Do the following steps:

- In the field for Connection name, define a Connection name.
- 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

The screenshots show the 'Add data source connection' dialog for Cloud Object Storage. The top dialog has fields for Manager API user, Manager API Password, Connection name (Test1), Connection type (Cloud Object Storage), and Manager host. The bottom dialog has fields for Manager host, Vault, Accessor Host, Accessor access key, Accessor secret key, and Site (optional).

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 above.

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 accesser 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 Spectrum® Discover graphical user interface (GUI), the metadata is transferred asynchronously back to the IBM Spectrum Discover instance.

Note:

IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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

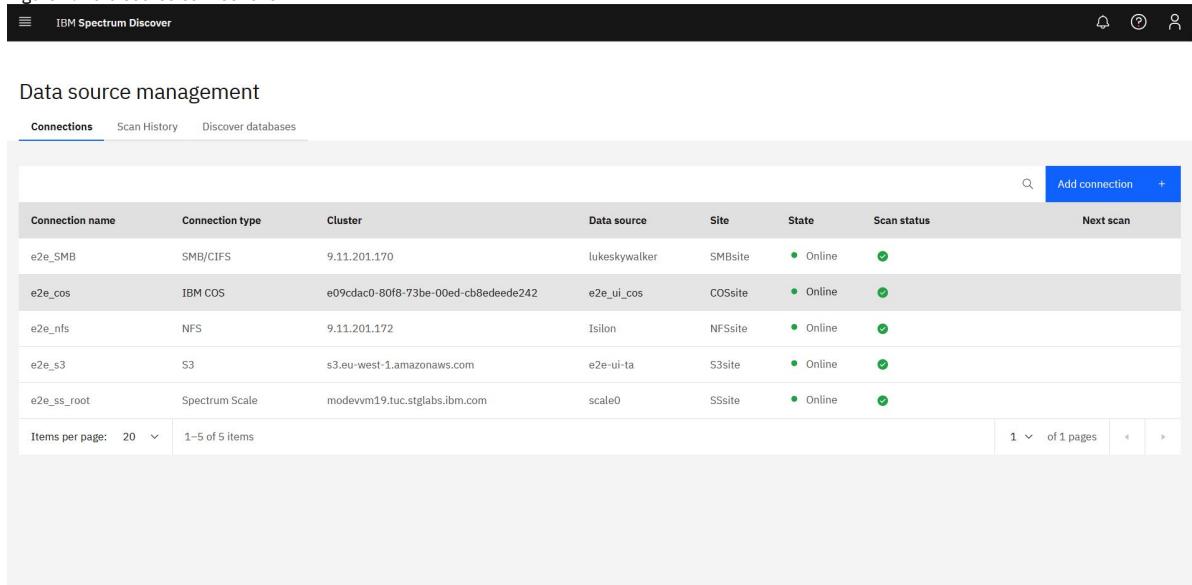
Procedure

1. Log in to the IBM Spectrum Discover 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



The screenshot shows the 'Data source management' page in the IBM Spectrum Discover GUI. The top navigation bar includes 'IBM Spectrum Discover', a search icon, and user profile icons. Below the header, there are three tabs: 'Connections' (which is selected), 'Scan History', and 'Discover databases'. A sub-header 'Data source management' is present. The main area is a table listing data source connections. The columns are: Connection name, Connection type, Cluster, Data source, Site, State, Scan status, and Next scan. The table contains five rows with the following data:

Connection name	Connection type	Cluster	Data source	Site	State	Scan status	Next scan
e2e_SMB	SMB/CIFS	9.11.201.170	lukeskywalker	SMBsite	● Online	●	
e2e_cos	IBM COS	e09cdac0-80f8-73be-00ed-cb8eedede242	e2e_ui_cos	COSsite	● Online	●	
e2e_nfs	NFS	9.11.201.172	Isilon	NFSSite	● Online	●	
e2e_s3	S3	s3.eu-west-1.amazonaws.com	e2e-ui-ta	S3site	● Online	●	

At the bottom of the table, there are pagination controls: 'Items per page: 20' and '1 of 1 pages'.

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 Spectrum Discover 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 Spectrum® Discover.

This file contains a list of all the messages that are not delivered to IBM Spectrum Discover.

/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 Spectrum Discover 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

Make sure you have the following features set up:

- IBM Spectrum® Discover 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 Spectrum Discover 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 Spectrum Discover.
2. To enable Ceph Object Storage bucket notifications:
 - a. Copy the ca.crt file from IBM Spectrum Discover node to a directory on the Ceph Object Gateway nodes.
 - b. Locate the file in the /etc/kafka directory on the IBM Spectrum Discover node.
 - c. Give this file a unique name on the Ceph node after it is copied over.
Remember: Make sure 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 Spectrum Discover where the notifications are sent to.
Remember: To enable notifications to be sent to IBM Spectrum Discover you must provide push endpoint parameters when you create the topic entity. These parameters include the IBM Spectrum Discover 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](#) 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 Spectrum Discover 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 Spectrum Discover node from the following location: /etc/kafka/sasl_password
<discover_fqdn>
    Indicates the fully qualified domain name of the IBM Spectrum Discover 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](#).

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

Make sure you have the following features set up:

- IBM Spectrum® Discover 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 Spectrum Discover 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 Spectrum Discover.

2. To enable Ceph Object Storage bucket notifications:

- a. Issue the following command to transfer the ca.crt file from IBM Spectrum Discover 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: Make sure 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 Spectrum Discover where the notifications are sent to.

Remember: To enable notifications to be sent to IBM Spectrum Discover, you must provide endpoint parameters when you create the topic entity.

These parameters include the IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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 Data Cataloging by using Kafka notifications.

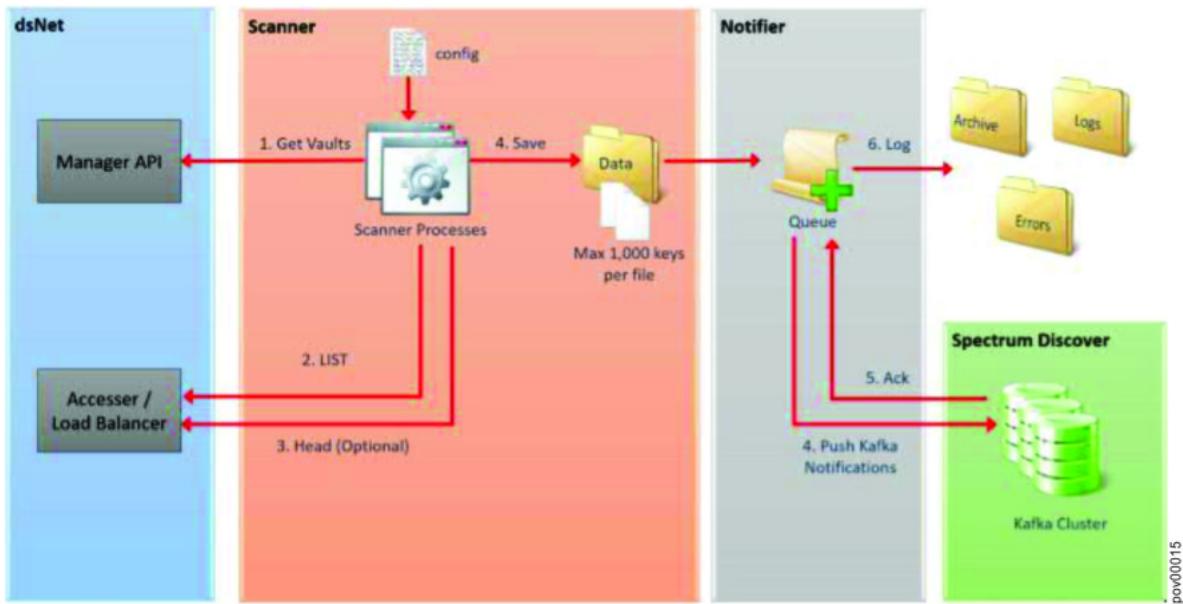
- [Overview of architecture](#)
This topic describes a high-level overview of IBM Cloud Object Storage Scanner architecture.
- [Configuration file](#)
The configuration file is used by the Notifier and Replay.
- [Replay performance](#)
The number of requests that are issued by IBM Cloud Object Storage 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 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 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 Spectrum® Discover 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					

Element	Description	Optional	Default value	Restart scanner if changed	Restart notifier if changed
System section					
min_utc	Only objects or version in the vaults that have a LastModified datetime on or after this timestamp is submitted to IBM Spectrum Discover. 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.	X		✓ See note.	X
max_utc	Only objects or version in the vaults that have a LastModified datetime on or before this time stamp is submitted to IBM Spectrum Discover. 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.	X
Policy engine section		(Only required for IBM Spectrum Discover 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.	X	none	✓	✓
user	Username for authorization on policy engine.	X	none	✓	✓
password	Password for authorization on policy engine.	X	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	X	✓
kafka_endpoint	IP address and port of the Kafka endpoint.	✓	Retrieved from Manager API	X	✓
kafka_topic	Name of the Kafka topic.	✓	Retrieved from Manager API	X	✓
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.	X	✓
kafka_password	The password for authentication with Kafka. Note: For testing only. Not to be used in production.	✓	Supplied by user at prompt if it cannot be retrieved from Manager API.	X	✓
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	X	✓
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	X	✓
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	✓	X
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	✓	X

Element	Description	Optional	Default value	Restart scanner if changed	Restart notifier if changed
System section					
<code>override_warnings</code>	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	✓	✗
<code>exclude_vaults</code>	Comma-separated list of vault names to be excluded from scanning, such as: "exclude-vaults": ["COSVault", "COSVault-V"]	✓	[] Empty list	✓	✗
<code>vaults</code>	List of vaults to be scanned. If <code>include_all_vaults</code> 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 <code>vault_name</code> is the name of the vault. The <code>has_custom_metadata</code> is an optional Boolean that overrides the <code>has_custom_metadata</code> that is described. The <code>prefix</code> 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.

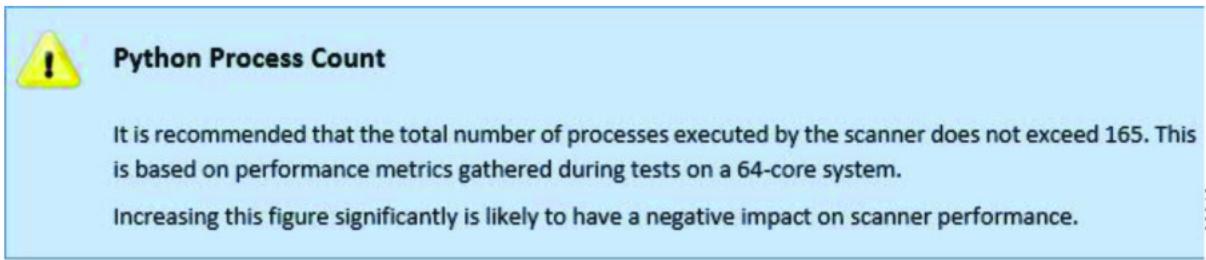
```
"replay": {
    "max_requests_per_second": 1000,
    "max_parallel_list": 10,
    "parallel_head_per_list": 15,
    "list_objects_size": 1000
}
```

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 sliceitors 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

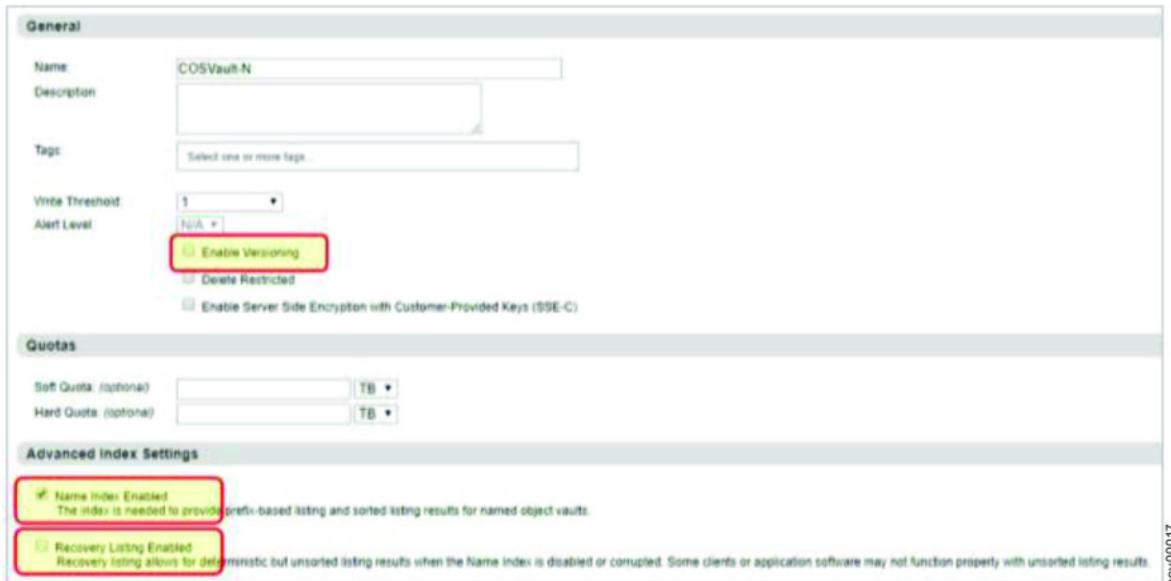
A few scenarios exist that prevent the Replay from operating correctly.

Certain combinations of the following IBM Cloud® Object Storage vault settings prevent the Replay from running a full scan:

- Vault versioning
- Name index
- Recovery 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	X	X	X	<input checked="" type="checkbox"/> 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: <code>"exclude-vaults": ["vault-name"]</code>
1	X	X	✓	<input checked="" type="checkbox"/> 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: <code>"exclude-vaults": ["vault-name"]</code>
2	X	✓	X	<input checked="" type="checkbox"/> Object Listing is run.
3	X	✓	✓	<input checked="" type="checkbox"/> 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 <code>override_warnings: true</code> 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	✓	X	X	<input checked="" type="checkbox"/> Object Listing is run.
5	✓	X	✓	<input checked="" type="checkbox"/> Object Listing is run.
6	✓	✓	X	<input checked="" type="checkbox"/> 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 <code>override-warnings: true</code> 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 Spectrum® Discover system.

Replay

When a severe outage occurs and causes the loss of notifications sent by the system to 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 access 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 Spectrum® Discover. 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 Spectrum® Discover, 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-c116cd33fa80	21,001	In progress	2018-08-01 11:03:48	13,000	13,000	13,000	0	0	62%	0%
Object	COSVault-NR	e:3e0eb6-7fd-7062-0143-3f59a1118180	3	Complete	2018-08-01 11:02:42	2	2	2	0	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	58%	46%
Version	COSVault-NV	6bd9011d-0d28-76fb-1132-dcc0540be380	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	b1bf437c-65b4-726a-108b-8c29a4065c80	20,001	In progress	2018-08-01 11:03:47	12,000	12,000	3,000	7,000	1,000	59%	58%
Object	EVO1_AWSV4_PERF1	43d01b33-ad4a-7d4b-1080-525024c3f980	101	Complete	2018-08-01 11:02:43	100	0	0	0	0	100%	100%
Object	EVO1_AWSV4_PLUGIN4	f355f8a7-461a-7aa4-10b0-5af47c519e80	1	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Object	EVO1_SMALL_VAULT1	a7fd6626-57c6-7359-1091-b5b7d5913b80	201	Complete	2018-08-01 11:02:43	200	0	0	0	0	100%	100%
Object	mega_vault?prefix=test1	97ecd99e-b7bb-7f1f-0198-ab16d346080	977,200	In progress	2018-08-01 10:51:31	107,000	107,000	80,000	27,000	0	10%	25%
Object	mega_vault?prefix=test2	97ecd99e-b7bb-7f1f-0198-ab16d346080	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-b7bb-7f1f-0198-ab16d346080	977,200	In progress	2018-08-01 10:51:32	109,000	109,000	61,000	48,000	0	11%	44%
Object	MGMTO001	a9114aba-4dcf-7bbd-10d4-f87f1c3ff380	14,187	Aborted	2018-08-01 11:03:46	13,000	9,701	9,701	0	0	91%	0%
Version	SuspendedVersioningTest-Vault	a5b5fce0-9018-7c0c-10b6-e50647064280	10	Complete	2018-08-01 11:02:42	9	9	9	0	0	100%	0%
Object	Threading-Test-2.1.5	91fb6b76-0ed9-767c-0153-c29101a74b80	0	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Object	Threading-Test-2.15.2	781e51bb-c9ee-7b8c-7115e-2f4ce53eab80	0	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Object	TimestampTest	65e10024-477e-7c99-11f7-6e93a09ce80	3	Complete	2018-08-01 11:02:42	2	2	2	0	0	100%	0%
Object	Vault-Empty	2892a6de-7ed4-736f-00da-a20debe4280	0	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Version	Vault-N	2344f296-a5af-7d0c-10d7-75acbcfd180	9	Complete	2018-08-01 11:02:42	8	8	8	0	0	100%	0%
Version	Vault-V	15ad56da-f304-77c3-00c5-f33ddff13480	12	Complete	2018-08-01 11:02:42	11	11	11	0	0	100%	0%
Object	vault1	c7b18026-873d-7e19-10ee-5b7b350ff880	0	Complete	2018-08-01 11:02:42	0	0	0	0	0	100%	100%
Version	version-delete-test	2db19e94-f498-7d8f-0135-1de35772280	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: <code>mega_vault?prefix=test</code>
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.
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 <code>/opt/ibm/metaocean/data/connections/cos/replay/output/data/<vault-name>/<prefix></code> directory. For each vault, see the <code>/opt/ibm/metaocean/data/connections/cos/replay/output/data/<vault-name>/<prefix></code> 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 <code>LastModified</code> 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 <code>notifier.debug</code> .
Approximate percentage scanned	Scanned as a percentage of Est. Object Count. The cell background shows a progress bar.

Column name	Description
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 Spectrum® Discover: <code>/opt/ibm/metaocean/data/connections/cos/replay/output/data</code>	<p>Contains .log files (Kafka messages), stats files and debug information for each scanned vault.</p>
For IBM Spectrum Discover: <code>/opt/ibm/metaocean/data/connections/cos/replay/output/debug/[scanner replay]/</code>	<p>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).</p>
For IBM Spectrum Discover: <code>/opt/ibm/metaocean/data/connections/cos/replay/output/debug/notifier</code>	<p>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.</p> <p>The notifier.debug rolls over when it reaches a predefined size as defined in the configuration file. See Configuration file.</p>
For IBM Spectrum Discover: <code>/opt/ibm/metaocean/data/connections/cos/replay/output/archive/</code>	<p>Contains all .log files that are successfully processed by the Notifier, and their contents are successfully submitted to the Kafka cluster.</p> <p>Remember: Files in this directory are truncated – they contain only the object key and they can also contain the version.</p>
For IBM Spectrum Discover: <code>/opt/ibm/metaocean/data/connections/cos/replay/output/error/</code>	<p>Contains any .log files that failed to submit to the Kafka cluster.</p>

Directory	Description
For IBM Spectrum Discover: <code>/opt/ibm/metaocean/data/connections/cos/replay/output/notification_log/</code>	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 <code>logging/notification_log_all</code> is true in the config file, all successful sends are also logged.

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 which 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

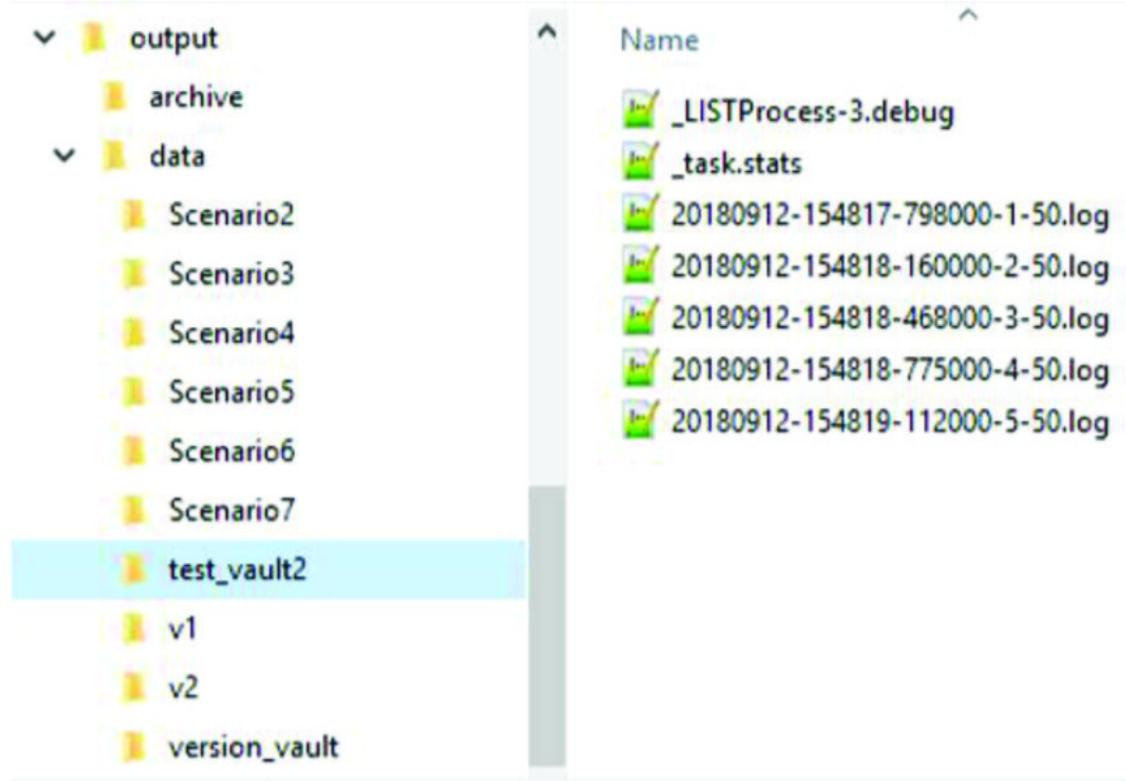


The status and progress of each scan must be maintained so a separate directory structure is created for each scan. [Table 1](#) shows the leaf directories that contain the file names and description.

Table 1. Leaf directory file names

File name	Description
-----------	-------------

File name	Description
_LISTProcessN.deb ug	<p>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.</p> <p>Figure 2. Example of running in debug mode</p> <pre data-bbox="414 255 1449 572"> 12-Sep-2018 15:58:13 LISTProcess-2 test_vault2 +++ Adding batch 16 12-Sep-2018 15:58:13 LISTProcess-2 test_vault2 16 Working batches: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16, 12-Sep-2018 15:58:13 LISTProcess-2 test_vault2 16 Working batches: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16, 12-Sep-2018 15:58:13 LISTProcess-2 test_vault2 --- Removing batch 1 12-Sep-2018 15:58:13 LISTProcess-2 test_vault2 Buffer is full. Sleeping for 1 second... 12-Sep-2018 15:58:14 LISTProcess-2 test_vault2 +++ Adding batch 17 12-Sep-2018 15:58:14 LISTProcess-2 test_vault2 16 Working batches: 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17, 12-Sep-2018 15:58:15 LISTProcess-2 test_vault2 16 Working batches: 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17, 12-Sep-2018 15:58:15 LISTProcess-2 test_vault2 --- Removing batch 2 12-Sep-2018 15:58:15 LISTProcess-2 test_vault2 +++ Adding batch 18 12-Sep-2018 15:58:15 LISTProcess-2 test_vault2 16 Working batches: 3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18, 12-Sep-2018 15:58:15 LISTProcess-2 test_vault2 16 Working batches: 3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18, 12-Sep-2018 15:58:15 LISTProcess-2 test_vault2 --- Removing batch 3 12-Sep-2018 15:58:15 LISTProcess-2 test_vault2 +++ Adding batch 19 12-Sep-2018 15:58:15 LISTProcess-2 test_vault2 16 Working batches: 4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19, 12-Sep-2018 15:58:15 LISTProcess-2 test_vault2 16 Working batches: 4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19, </pre> <p style="text-align: right;">. pov00006</p>
task.stats	<p>Scanner starts in JSON format for a single vault. Updated following successful processing of each batch of objects.</p> <pre data-bbox="372 635 1351 1184"> "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-411b-c7550651a780" </pre> <p style="text-align: right;">. pov00007</p>

File name	Description
*.log	<p>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.</p> <p>The naming convention for the log files is</p> <pre data-bbox="344 219 894 261"><date>-<time>-<milliseconds>-<batch number>-<number of messages in file>.log</pre> 

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

pov00008

```

("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-dc7fec7b8e1", "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

```

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 |     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 |     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 |     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 |     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-eqx204201-
mgr-03
12-Sep-2018 15:57:28 |         |- device_id:2    accesser   172.19.17.39  39a14ec8-090c-79d8-10f9-255249197f43vm-eqx204201-

```

pov00010

Figure 3. Scanner debug (continued)

```

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 |     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 |     Retrieving vault information from dsNet
12-Sep-2018 15:57:26 | -----
12-Sep-2018 15:57:27 |     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 |     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 |     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 15:57:27 |         |- 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-eqx204201-
12-Sep-2018 15:57:28 |             |- device_id:2    accesser   172.19.17.39  39a14ec8-090c-79d8-10f9-255249197f43vm-eqx204201-

```

psv000011

Figure 4. Scanner debug (continued)

```

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 |     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-eqx204201-
mgr-03
12-Sep-2018 15:57:28 |     | - device_id:2    accesser   172.19.17.39  39a14ec8-090c-79d8-10f9-255249197f43vm-eqx204201-

```

pov00012

Figure 5. Scanner debug (continued)

```
12-Sep-2018 15:57:29 | 10 tasks
12-Sep-2018 15:57:29 |
12-Sep-2018 15:57:29 | |- object scan of v1
12-Sep-2018 15:57:29 |   |- Throttling list: 0.0500 seconds, Head: 0.0050 seconds
12-Sep-2018 15:57:29 | |- Object Scan of Scenario3
12-Sep-2018 15:57:29 |   |- Throttling list: 0.0500 seconds, Head: 0.0050 seconds
12-Sep-2018 15:57:29 | |- object Scan of Scenario2
12-Sep-2018 15:57:29 |   |- Throttling List: 0.0500 seconds, Head: 0.0050 seconds
12-Sep-2018 15:57:29 | |- object Scan of v2
12-Sep-2018 15:57:29 |   |- Throttling list: 0.0500 seconds, Head: 0.0050 seconds
12-Sep-2018 15:57:29 | |- Version Scan of version_vault
12-Sep-2018 15:57:29 |   |- Throttling list: 0.0010 seconds, Head: n/a
12-Sep-2018 15:57:29 | |- Object Scan of Scenario6
12-Sep-2018 15:57:29 |   |- Throttling list: 0.0500 seconds, Head: 0.0050 seconds
12-Sep-2018 15:57:29 | |- Version Scan of Scenario5
12-Sep-2018 15:57:29 |   |- Throttling list: 0.0010 seconds, Head: n/a
12-Sep-2018 15:57:29 | |- Object Scan of Scenario4
12-Sep-2018 15:57:29 |   |- Throttling list: 0.0500 seconds, Head: 0.0050 seconds
12-Sep-2018 15:57:29 | |- Object Scan of Scenario7
12-Sep-2018 15:57:29 |   |- Throttling list: 0.0500 seconds, Head: 0.0050 seconds
12-Sep-2018 15:57:29 | |- Object Scan of test_vault2
12-Sep-2018 15:57:29 |   |- Throttling list: 0.0500 seconds, Head: 0.0050 seconds

12-Sep-2018 15:57:29 | -----
12-Sep-2018 15:57:29 | Ignoring vaults
12-Sep-2018 15:57:29 | -----
12-Sep-2018 15:57:29 | |- Scenario1
12-Sep-2018 15:57:29 | |- Scenario0

12-Sep-2018 15:57:29 | -----
12-Sep-2018 15:57:29 | Queuing scanner tasks
12-Sep-2018 15:57:29 | -----
12-Sep-2018 15:57:29 | |- Queuing task 'Object Scan of v1'
12-Sep-2018 15:57:29 | |- Queuing task 'Object Scan of Scenario3'
12-Sep-2018 15:57:29 | |- Queuing task 'Object Scan of Scenario2'
12-Sep-2018 15:57:29 | |- Queuing task 'object Scan of v2'
12-Sep-2018 15:57:29 | |- Queuing task 'Version Scan of version_vault'
12-Sep-2018 15:57:29 | |- Queuing task 'Object Scan of Scenario6'
12-Sep-2018 15:57:29 | |- Queuing task 'Version Scan of Scenario5'
12-Sep-2018 15:57:29 | |- Queuing task 'Object Scan of Scenario4'
12-Sep-2018 15:57:29 | |- Queuing task 'Object Scan of Scenario7'
12-Sep-2018 15:57:29 | |- Queuing task 'Object Scan of test_vault2'

12-Sep-2018 15:57:29 | -----
12-Sep-2018 15:57:29 | Creating 10 list processes, each with 5 head processes
12-Sep-2018 15:57:29 | -----
12-Sep-2018 15:57:31 | |- Started LISTProcess-0
12-Sep-2018 15:57:32 |   |- Started HEADProcess-0-0
12-Sep-2018 15:57:33 |   |- Started HEADProcess-0-1
12-Sep-2018 15:57:34 |   |- Started HEADProcess-0-2
12-Sep-2018 15:57:35 |   |- Started HEADProcess-0-3
12-Sep-2018 15:57:36 |   |- Started HEADProcess-0-4
12-Sep-2018 15:57:38 |   |- Started LISTProcess-1
12-Sep-2018 15:57:39 |     |- Started HEADProcess-1-0
12-Sep-2018 15:57:40 |     |- Started HEADProcess-1-1
12-Sep-2018 15:57:41 |     |- Started HEADProcess-1-2
12-Sep-2018 15:57:42 |     |- Started HEADProcess-1-3
12-Sep-2018 15:57:43 |     |- Started HEADProcess-1-4
12-Sep-2018 15:57:44 |   |- Started LISTProcess-2
12-Sep-2018 15:57:46 |     |- Started HEADProcess-2-0
12-Sep-2018 15:57:47 |     |- Started HEADProcess-2-1
12-Sep-2018 15:57:48 |     |- Started HEADProcess-2-2
12-Sep-2018 15:57:50 |     |- Started HEADProcess-2-3
12-Sep-2018 15:57:52 |     |- Started HEADProcess-2-4
12-Sep-2018 15:57:54 |     |- Started LISTProcess-3
```

nov00013

Configure IBM Cloud Object Storage notifications for Data Cataloging

Ingesting IBM Cloud® Object Storage event records into 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 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 Data Cataloging cluster. The certificate and credentials required to establish this connection might be obtained directly from the Data Cataloging cluster by the Data Cataloging storage administrator.

For information on how to enable and configure the IBM Cloud Object Storage Notification service with the 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 Data Cataloging:

Hosts

One or more of the 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 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 Data Cataloging storage administrator from the following location on the Data Cataloging master node: */etc/kafka/server.properties*.

Authentication credentials

The username is **cos** and the password can be obtained by the Data Cataloging storage administrator from the following location on the 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 Data Cataloging cluster. It might be obtained by the Data Cataloging storage administrator from the following location on the 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 Spectrum® Discover 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 Spectrum Discover 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 Spectrum Discover platform. After you collect the information, you can add the information to the notification service configuration.

- [Authenticating, encrypting, and enabling](#)
- [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

- Log in to the IBM Spectrum® Discover 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 = meezeDMxFNZJMSxdyWQKSjVbs
```

Encryption

The following information shows an example of a certificate of authority for the PEM file.

1. Log in to the IBM Spectrum Discover 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-----
MIIExTCCA62gWBIBAgJIAKJM/x6ULb6YMA0GCSqSjSIB3DQEBCwUAMIGYMQswCQYD
QQGEWbJHQjeOMAwGA1UECAwFSEFOVFwxEADOBgNVBACmB0h1cnNsZXkxDDAKBgNV
BAoMA01CTTEZMbcGA1UECwQzC3B1Y3RydWlkaxNjb3Z1cjEZMBCGA1UEAwQc3B1
Y3RydWlkaxNjb3Z1cjEjMCEGCSqSjSIB3DQEJARYWbXhd3J1bmN1QHVRlmlb5sj
b20wHncNMtkwMTAyMTY1MDU5WhCNzgxMjI4MTY1MDU5WjCBnDELM4kGA1UEBh
R0IxJdAMBvNVAQBgMBuB1TlRTMRwAsgDyVQDQHAd1dxJzBgv5MqwCgYDVQCDANJ
Qk0xGTAXBgNVBASMEHNwZWN0cnVtZG1zY292ZXIxGTAXBgNVBAMMEHNwZWN0cnVt
ZG1zY292ZXIxLzAhBgkqhkiG9w0BCWEFG1sYXdyZw5jUB1ay5pYm0uY29tMIB
IjANBkgkqhkiG9w0BQAeFAOAQCA8AM1BkgCKAQEAwg7z4gdEfWk1eJpvj3wobD
JrHJngooDbPLicRSF/yj11NgwbWbjIj1eL9R8My+24hRUGfym9IWCM8qMwHEHg+w
+Rr+6dQdy89j+m1c21y3nKhxyTsQZQr03Uy1C/timF6c0T7CfuQ1E21jh/H/JXVK4
ESViLhZR23+fdPbITZLmLvdftJxs0Kgu0w4B1R9kpQ3bXwt+eoAvhdKztD
1YLCmdzf0i6E3aspxRhcsGW3bCmu5qzT6E8BnsSxrZkRbdL6Q0Pqv33XVxP6z
OHIVv1uFg9Vq6XH1ZLBhWNDqPgYoAbT0Q43vUxx7mJ3uJQY6gbfbuFa+PxygQwID
AQAB4IBDjCCA0qHWDYRVOOBHBFEEFKmmHesFsfhgHuFL1dd82WMyf190YQ10MHNBgNV
HSMEGucWgkCFKEKmmHesFsfhgHuFL1dd82WMyf190YQ10HgMIGYMQswCQYDVQD
EwJHQjeOMAwGA1UECAwFSEFOVFwxEADOBgNVBACmB0h1cnNsZXkxDDAKBgNVBAoM
A01CTTEZMbcGA1UECwQzC3B1Y3RydWlkaxNjb3Z1cjEZMBCGA1UEAwQc3B1Y3Ry
dWlkaxNjb3Z1cjEjMCEGCSqSjSIB3DQEJARYWbXhd3J1bmN1QHVRlmlb5jB2C
CQjF+5+1c2+dAMBdgNVHRMEBTADAQH/MAsGa1UdwQeAWBjB1anBkgkqhkiG9w0B
AqsFAAOCAQEANIRvyeuJh69irK5dPJssmcISXcZv4X33ukAyRt4zLNFToSktfj2
ZatCQNCng91Ln7Twu1te+6wifxka&UD7wrxMzb32+Mpw/XNz05DnhInfvkAF62
SHqWiaqtTLXdeG807ieFisI7kAgEOfC23/z+BSw+2m1XB1UcxuMioYwX4YTb14/
DLJkqkhXMLWv+h/7NU7KbeRSBia24N5z1R6Ed/rx83uD2AwBnBq7t24sD6Q8Gbmt
HLMyvJrH1vtly1vgSfkZnShb+E6V/5+GsnpladylpsCvM1LqS/wMzb9g1hT5si181
nmqmTK6yqccS7CfwFw/DjQr/19EcYj8fAQ==
-----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

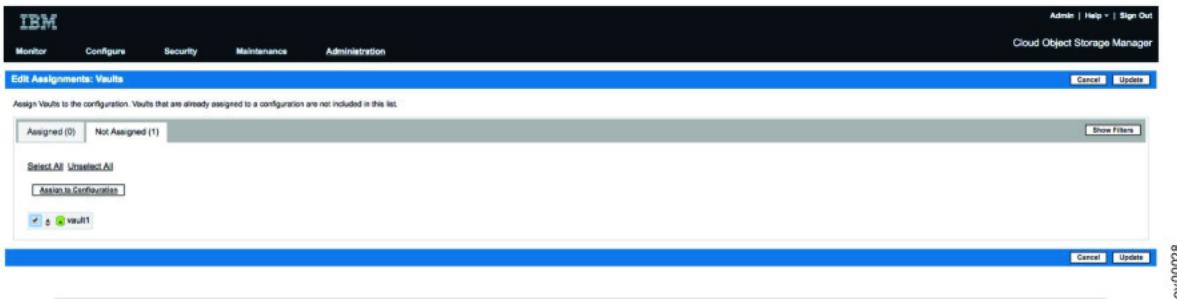
- Check Enable authentication.

Username: cos
Password: <PASSWORD>

Enabling encryption

1. Check Enable TLS for Apache Kafka network connections.
 2. Add the certificate PEM file from the IBM Spectrum Discover 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 Spectrum® Discover server and issue the following command to extract the password information:

as set `secret.kafka.ssl.password`, `libm-data-cataloging`, `isopathn1`'s `data.password`).

Encryption

The following information shows an example of a certificate of authority for the PEM file:

1. Log in to the IBM Spectrum Discover 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  
VQOGEWjHQjROMAwGA1UECAwFSEFOVFMxEDAOBgNVBAcMB0h1cnNsZXkxDAAKbgNV  
BAoMA01CTTEZMBcGA1UECwwQc3B1Y3RydW1kaXNjb3ZlcjEZMBcGA1UEAwQc3B1  
Y3RydW1kaXNjb3ZlcjEjMCEGCSqGSIb3DQEJARYUbWxhd3JlbnNlQHvLmlibS5j  
b20wHhcNMTkWMTAyMTU1MDU5WhcNMQzgxMjI4MTY1MDU5WjCBmDELMakGA1UEBhMC  
R0IxJdJAMBgNVBAgQMBUGhBT1RTMRawDgYDVQQHDAIdXJzbGV5MQwwCgYDVQQKDANJ  
Qk0xGTAXBgNVBAsMEHNwZWN0cnVtZG1zY292ZXIxGTAXBgNVBAMMEHNwZWN0cnVt  
ZG1zY292ZXIxIzAhBgkhkiG9w0BCQEWFg1sYXdyZw5jZUB1ay5pYm0uY29tMIIB  
IjANBgkqhkiG9w0BAQEFAOAQ8AMIBEcqKCAQEawg7z4gdewlkeJjPvj3wobDBB  
JrHJngooDbPLicRSF/yj11NgwbWbjIjIeL9R8My+24hRUGfym9IWCM8gMWyEHG+w  
+Rx/6jdQyD9j+m1c2ly3ndhXYsTQZR03Uy1C/TimF6fc07cfuQ1E21jHf/JXVK4  
ESVi1hZR23/tWIfbITZmLvdftJSx0Kgu0Ow4B1r9kpQ3bXwt/edovAhdKztDowWN  
LYCGmdzFOiE63asspxHchsGW3bcMu5mqzT6BEoSzrrx8kRbRDL6Q0Pqv33XVxP6z  
OHIVvluFg9v6KHIZLBhWNDqPgYoAbT0Q43vUxk7mJ3uJQY6gbfUea+PxxygQwID  
AQABo4IBDjCCAQowHQYDVR0BByEFExmmHeSfxgHuFL1dd82WMyf190MIHNBgNV  
HSMEgcUwgcrKAFeKxmmHeSfxgHuFL1dd82WMyf190oYGeplGmIGYMQswCQYDVQQG  
EwJHQjEOMAwGA1UECAwFSEFOVFMxEDAOBgNVBAcMB0h1cnNsZXkxDAAKbgNVBAoM  
A01CTTEZMBcGA1UECwwQc3B1Y3RydW1kaXNjb3ZlcjEZMBcGA1UEAwQc3B1Y3Ry  
dw1kaXNjb3ZlcjEjMCEGCSqGSIb3DQEJARYUbWxhd3JlbnNlQHvLmlibS5j22C  
CQCjF/5+1C2+mDAMBgNVHRMEBTADAQH/MAsgA1UdDwQEAwIBBjANBgkqhkiG9w0B  
AQsFAAACQEAQANINRvyeeuJh69iRK5dPJssmcISXcZv4X33ukAyRt4zLNFToSktfj2  
ZAtQCNqN19Ln7Twuit+e6wifxAkA+UD7wrzMzb32+Mpw/XNzo5DnhInfvkAfc62  
SHqWIaqTLDDeGBe807ieFs17kAgEQCF2z/vesB2+m1XB1l0cuxMiOywX4Ytb14/  
GLDJkqhXMLWVh+/7NU7KbERSBia24Ns2lR6Ed/rx83ud2AwBnBqt24sD6Q8Gbme  
HILMv0JrH1vty1vGsfkZnShb+E6V/5+GsnpiAaDyIpsCvM1LqS/wMzB9h1T5sii81  
mmqMTK6yqcs7CFWFV/DjQz/i9EcjJ8FAQ==  
-----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

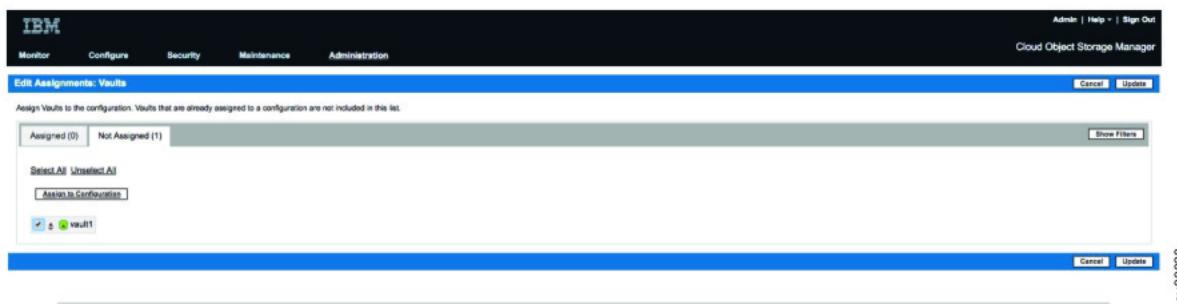
- Check Enable authentication.

```
Username: cos
Password: <PASSWORD>
```

Enabling encryption

1. Check Enable TLS for Apache Kafka network connections.
2. Add the certificate PEM file from the IBM Spectrum Discover 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 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 Data Cataloging server, the accesser logs can be monitored on the IBM Cloud Object Storage Accesser server.
- **Monitoring the Data Cataloging producer IBM Cloud Object Storage logs**
When the Data Cataloging server receives a notification from the IBM Cloud Object Storage platform, the Data Cataloging producer IBM Cloud Object Storage records a transaction.
- **Monitoring the IBM Spectrum Discover dashboard for IBM Cloud Object Storage ingestion**
You can monitor the IBM Spectrum® Discover 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 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 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 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 Data Cataloging producer IBM Cloud Object Storage logs

When the Data Cataloging server receives a notification from the IBM Cloud® Object Storage platform, the 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@spectrum_discover]$ 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}]
```

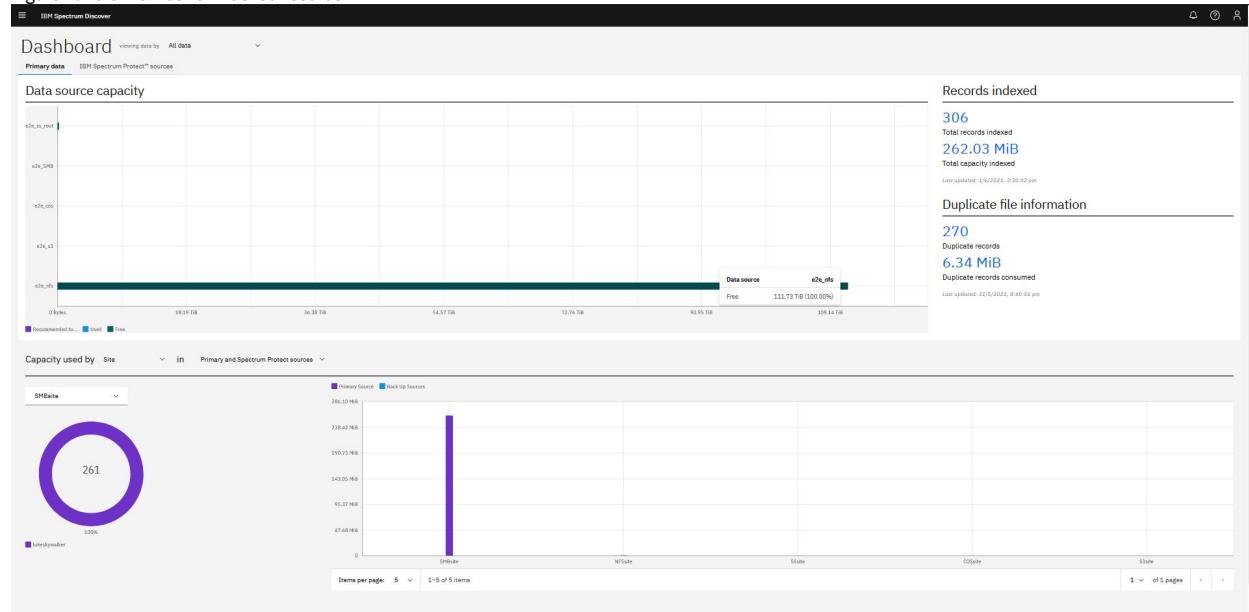
Monitoring the IBM Spectrum Discover dashboard for IBM Cloud Object Storage ingestion

You can monitor the IBM Spectrum® Discover dashboard for IBM Cloud® Object Storage ingestion.

After IBM Cloud Object Storage notifications are ingested from the IBM Cloud Object Storage platform, the IBM Spectrum Discover dashboard displays the total number of indexed records.

Note: The IBM Spectrum Discover dashboard can take approximately 30 minutes to display the total number of indexed records. See [Figure 1](#).

Figure 1. Total number of indexed records



Data Cataloging and S3 object storage data source connections

Use this information to understand how 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

- Log in to the IBM Spectrum® Discover 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](#).
- 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

e2e_SMB selected							
Connection name	Connection type	Cluster	Data source	Site	State	Scan status	Next scan
e2e_SMB	SMB/CIFS	9.11.201.170	lukeskywalker	SMBsite	● Online	●	
e2e_cos	IBM COS	e09cdac0-80f8-73be-00ed-cb8ede0e242	e2e_ui_cos	COSSite	● Online	●	
e2e_nfs	NFS	9.11.201.172	Isilon	NFSsite	● Online	●	
e2e_s3	S3	s3.eu-west-1.amazonaws.com	e2e-ui-ta	S3site	● Online	●	
e2e_ss_root	Spectrum Scale	modevwm19.tuc.stqlabs.ibm.com	scale0	SSsite	● Online	●	

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

Add data source connection

Connection name
Connection name *
Field is required

Connection type
Choose an option

- Cloud Object Storage
- Network File System
- Spectrum Scale
- Simple Storage Service (S3)**
- IBM Spectrum Protect
- Server Message Block (SMB) / CIFS

Cancel Submit Connection

4. Complete the following steps:
- In the field for Connection name, define a Connection name.
 - 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)

Add data source connection

Connection name
Test_scale

Connection type
Choose an option

- Cloud Object Storage
- Network File System
- Spectrum Scale
- Simple Storage Service (S3)**
- IBM Spectrum Protect
- Server Message Block (SMB) / CIFS

Cancel Submit Connection

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

The screenshot shows the 'Add data source connection' dialog for Simple Storage Service (S3). The 'Connection name' field is highlighted with a red border and has a red exclamation mark icon indicating it is required. Other fields include 'Access key', 'Instance user password', 'Site (optional)', 'Storage host', and 'Bucket name'. There are also checkboxes for 'Select a collection' and 'Schedule data scans'. At the bottom, there are 'Cancel' and 'Submit Connection' buttons.

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 Spectrum® Discover graphical user interface (GUI), the metadata is transferred asynchronously back to the IBM Spectrum Discover instance.

Automated scanning and data ingestion relies on an established and active network connection between the IBM Spectrum Discover 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 Spectrum Discover GUI for that connection.

Procedure

1. Log in to IBM Spectrum Discover web interface.
 2. Click  go to Data connections > Connections.
- The following figure shows the data connections lists:

Figure 1. Data source connections

Connection name	Connection type	Cluster	Data source	Site	State	Scan status	Next scan
e2e_SMB	SMB/CIFS	9.11.201.170	lukeskywalker	SMBsite	● Online	●	
e2e_cos	IBM COS	e09cdac0-80f8-73be-00ed-cb8e0ede242	e2e_ui_cos	COSsite	● Online	●	
e2e_nfs	NFS	9.11.201.172	Isilon	NFSSite	● Online	●	
e2e_s3	S3	s3.eu-west-1.amazonaws.com	e2e-ui-ta	S3site	● Online	●	
e2e_ss_root	Spectrum Scale	modevmm19.tuc.stglabs.ibm.com	scale0	SSsite	● Online	●	

3. Select the data source connection name that you want to scan. Make sure 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.

Connection name	Connection type	Cluster	Data source	Site	State	Scan status	Next scan
e2e_SMB	SMB/CIFS	9.11.201.170	lukeskywalker	SMBsite	● Online	●	
e2e_cos	IBM COS	e09cdac0-80f8-73be-00ed-cb8e0ede242	e2e_ui_cos	COSsite	● Online	●	
e2e_nfs	NFS	9.11.201.172	Isilon	NFSSite	● Online	●	
e2e_s3	S3	s3.eu-west-1.amazonaws.com	e2e-ui-ta	S3site	● Online	●	
e2e_ss_root	Spectrum Scale	modevmm19.tuc.stglabs.ibm.com	scale0	SSsite	● Online	●	

4. Select Scan now to change the status to Scanning.
The following figure shows an active scan.

Figure 3. Active scans

Connection name	Connection type	Cluster	Data source	Site	State	Scan status	Next scan
e2e_SMB	SMB/CIFS	9.11.201.170	lukeskywalker	SMBsite	● Online	Scanning SMB mount (Step 2 of 3) Scanned: 0 Indexed: 0	
e2e_cos	IBM COS	e09cdac0-80f8-73be-00ed-cb8ede0e242	e2e_ui_cos	COSSite	● Online	✓	
e2e_nfs	NFS	9.11.201.172	Isilon	NFSSite	● Online	✓	
e2e_s3	S3	s3.eu-west-1.amazonaws.com	e2e-ui-ta	S3site	● Online	✓	
e2e_ss_root	Spectrum Scale	modevmm19.tuc.stglabs.ibm.com	scale0	SSsite	● Online	✓	

- 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 Spectrum® Discover.

/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 Spectrum Discover 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 Spectrum® Discover GUI to scan an Elastic Storage Server (ESS) data connection.

Procedure

- Log in to IBM Spectrum Discover web interface with data admin privileges.
- Open an internet browser to the IP address or host name of your IBM Spectrum Discover cluster.
The default credentials to put into the IBM Spectrum Discover dialog box are:

```
Default user: sdadmin
Default Password: Passw0rd
```

- Click menu and go to Data connections > Add data source connection.
- 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 Spectrum® Discover 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 Spectrum Discover identifies source storage systems that are to be indexed by IBM Spectrum Discover.

For some data source types, a network connection can be created to allow for automated scanning and indexing of the source system metadata. IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover.

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 Spectrum® Discover graphical user interface.

About this task

Create data source connections in IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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

The name of the connection, an identifier for the user, for example Share1.

Note: It must be a unique name within IBM Spectrum Discover.

Share

The path name of the SMB/CIFS file share, for example //server/share.

User

The user ID that has access permissions to the SMB share, for example DOMAIN1\user1

Password

The authentication password for the user ID provided.

Site (Optional)

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 Spectrum® Discover graphical user interface.

About this task

Creating data source connections in IBM Spectrum Discover identifies source storage systems that are to be indexed by IBM Spectrum Discover.

For some data source types, a network connection is (optionally) created to allow for automated scanning and indexing of the source system metadata. IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover.

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 Spectrum® Discover system through offline mode.

Before you begin

- You must have the `SD_USER`, `SD_PASSWORD` & `OVA` environment variables set in the IBM Spectrum Discover system. To verify the details, enter `gettoken`.
- Enter the connection values inside the JSON file and add it to IBM Spectrum Discover 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 Spectrum Discover host system with following credentials.

```
Host: <IBM Spectrum  
Discover_hostname>  
login: moadmin  
Password: PasswOrd
```

2. SCP the list.metaocean file to the IBM Spectrum Discover 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 Spectrum Discover 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 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 - 1TB	L
1TB+	XL

After you change or update a bucket definition, IBM Spectrum® Discover 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

Modify Bucket

Bucket Name

SizeRange

extra small

less than 4 KiB

small

4 KiB through 1 MiB

medium

1 MiB through

large

Cancel Submit

pov00054

The screenshot shows a 'Modify Bucket' interface with a 'SizeRange' section. It includes dropdown menus for selecting ranges: 'less than' (set to 4 KiB) and 'through' (set to 1 MiB). Other options like 'extra small', 'small', 'medium', and 'large' are also listed. At the bottom are 'Cancel' and 'Submit' buttons, with the identifier 'pov00054' to the right.

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

Bucket Name
TimeSinceAccess

- 1 year+
- more than 1 year
- 1 quarter
- 1 month through
- 1 year
- 3 months through
- 1 month

Cancel **Submit**

pov0055

Using custom TLS certificate

You can change the TLS certificate that is used by Data Cataloging for serving web pages and the REST API endpoints.

About this task

Follow the procedure to use a custom TLS certificate:

Procedure

- Create a secret for your TLS certificate within the same namespace as the one used for deploying Data Cataloging on OpenShift® that is "spectrum-discover". Note: You can use any name for the secret. The following example uses "my-tls-secret" as the secret name.

```
oc create secret tls my-tls-secret --key ${KEY_FILE} --cert ${CERT_FILE} -n ibm-data-cataloging
```

- Modify the Data Cataloging custom resource and specify the following ingress settings:

```
oc edit SpectrumDiscover spectrumdiscover -n ibm-data-cataloging
```

- Update the "host" and "tls_secret_name" in the relevant ingress section.

```
ingress:
  host: spectrum-discover.ibm.com
  tls_secret_name: my-tls-secret
```

Note: The "ingress.host" setting must match the fully qualified domain name as specified in the TLS certificate. This domain name is the hostname that the ingress binds to.

- Save the custom resource.

Note: The operator takes a while to go through all components and update them with the new settings. Issue the following command to check the operator log for monitoring its progress.

```
oc logs $(oc get po -l name=spectrum-discover-operator -n ibm-data-cataloging -o yaml) -n ibm-data-cataloging -c operator --follow
```

- The log displays "PLAY RECAP" on completing the update.
- Enter **ctrl+c** to stop following the log.

IBM Spectrum Storage software requirements

Use Data Cataloging to index metadata from other applications and to orchestrate the data management.

IBM Cloud Object Storage

Data Cataloging indexes metadata from IBM Cloud® Object Storage by receiving notifications that contain metadata from IBM Cloud Object Storage. 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 Data Cataloging:

Table 1. IBM Cloud Object Storage software requirements

Component	Version
IBM Cloud Object Storage	3.14.0 and higher

IBM Storage Scale

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 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 Spectrum® Discover 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

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 Data Cataloging:

Table 3. IBM Spectrum Protect software requirements

Component	Version
IBM Storage Protect	7.x and higher

IBM Storage Archive

Data Cataloging supports the advanced tiering function with the ScaleILM application.

The following table lists the minimum IBM Storage Archive software version that is required to control the data placement by Data Cataloging.

Table 4. IBM Storage Archive software requirements

Component	Version
IBM Storage Archive Enterprise Edition (EE)	1.3.0.6 and higher

Data Cataloging installation with alternative VLAN

Install and configure Data Cataloging that uses additional VLAN connected through the IBM Storage Fusion upstream links.

For security reasons, the target VLAN network is not available for the IBM Storage Fusion cluster through the default IBM Storage Fusion public network.

There are several approaches available to access alternate VLANs that do not have routable access through the default IBM Storage Fusion network.

The remainder of this document covers the option of OpenShift® **NetworkAttachmentDefinitions** to provide Data Cataloging access to alternative VLAN access. It explains the setup and implementation of **NetworkAttachmentDefinitions** and the required modifications to the Data Cataloging definitions.

IBM Storage 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.

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 Storage 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: Once 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 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 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 Data Cataloging service to the running state. For the procedure, see [Returning Data Cataloging to a running state](#).

Example Scaling Commands:

- Scale down 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 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 00FFFFFF 0 0 0  
br4001 0064640A 0164640A 0003 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  
|-- 10.100.0.0  
|-- 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 Data Cataloging data connection. For the procedure, see [Creating an IBM Storage Scale data source connection](#).

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 via 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 Spectrum Discover, 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 Spectrum Discover 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 Spectrum Discover. 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 Spectrum Discover. 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 Spectrum Discover users.

Initial login

To log in IBM Spectrum® Discover graphical user interface, use the following information.

IBM Spectrum Discover 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 Spectrum Discover user interface 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 Spectrum Discover graphical user interface:

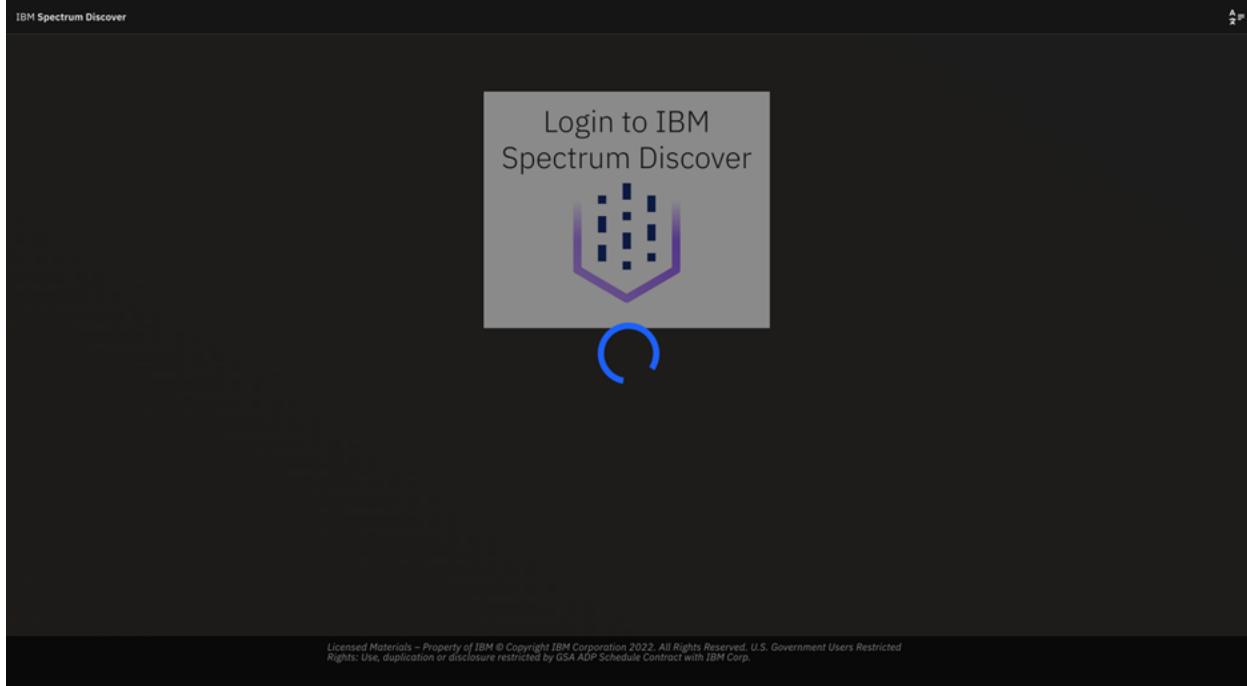
Username
sdadmin
Password
Passw0rd

Important: When the SSO is disabled, the administrator must change the password during the initial login.

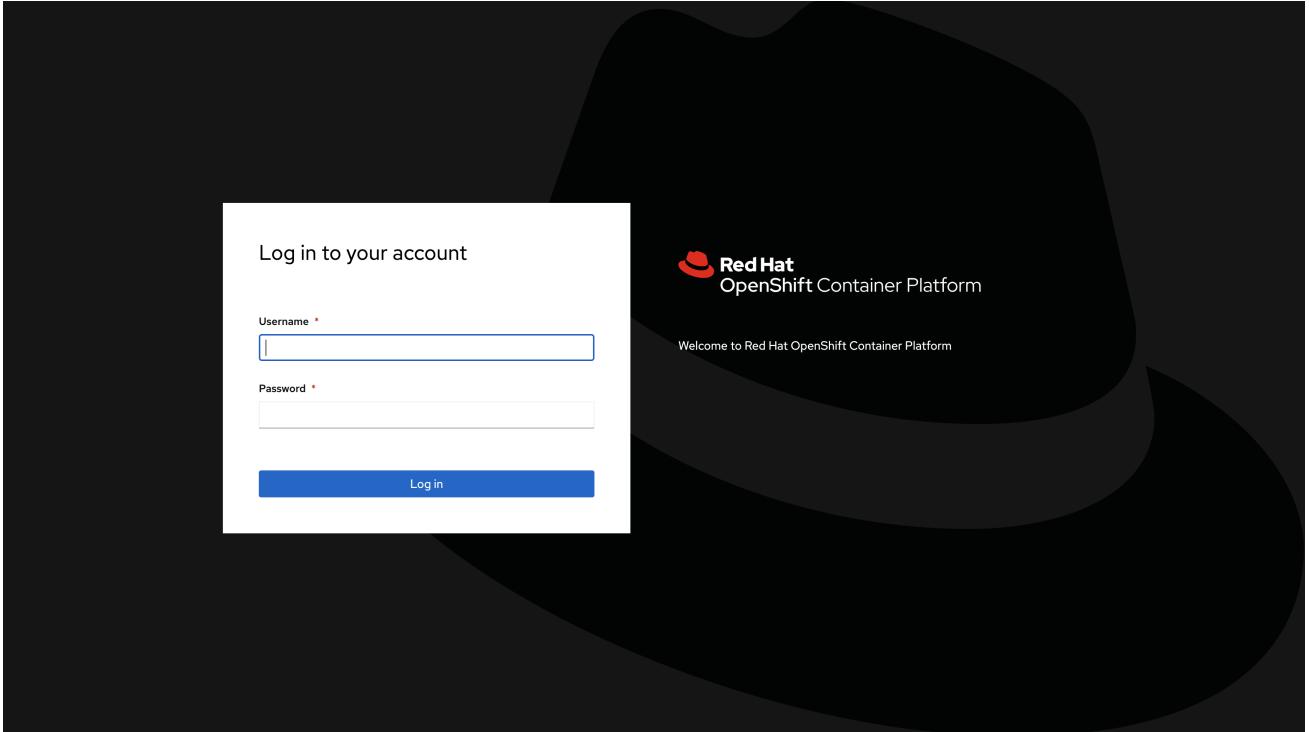
Log in to IBM Spectrum Discover Dashboard with SSO enabled

Follow the steps to log in to the IBM Spectrum Discover Dashboard:

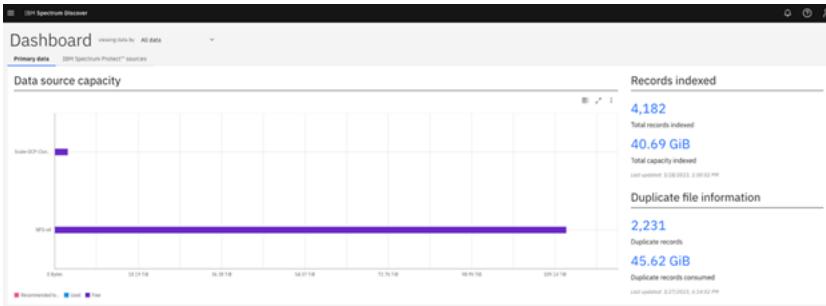
1. Launch the IBM Spectrum Discover graphical user interface, and you get the logging screen validation as follows.



2. Wait for the validation for OpenShift Platform session. If you have an active session, you will be redirected to Data Cataloging Dashboard, otherwise you will be redirected to the OpenShift authentication client.



3. Enter your OpenShift credentials and click Log in button.
You will be redirected to IBM Spectrum Discover main dashboard.

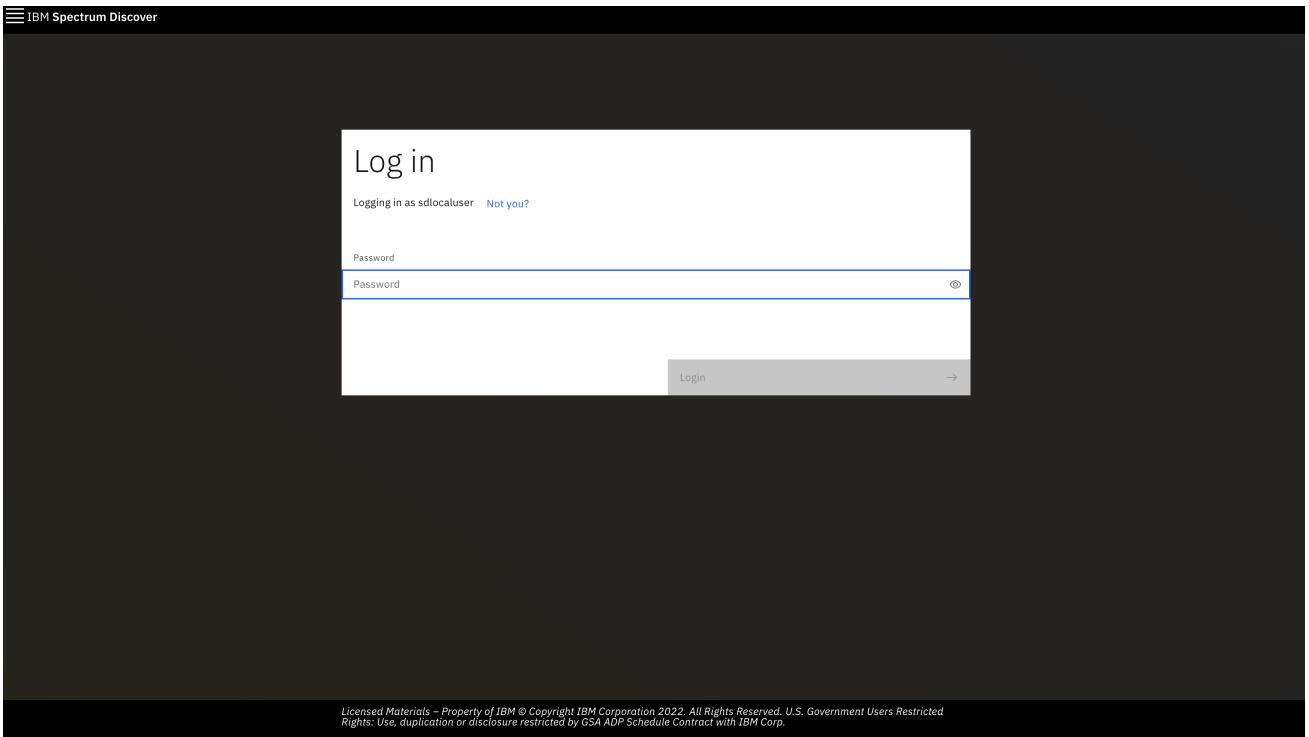


Note: Consider that IBM Spectrum Discover for security renews the session every hour, and there are cases in which it will be necessary to re-enter the OpenShift Container Platform credentials.

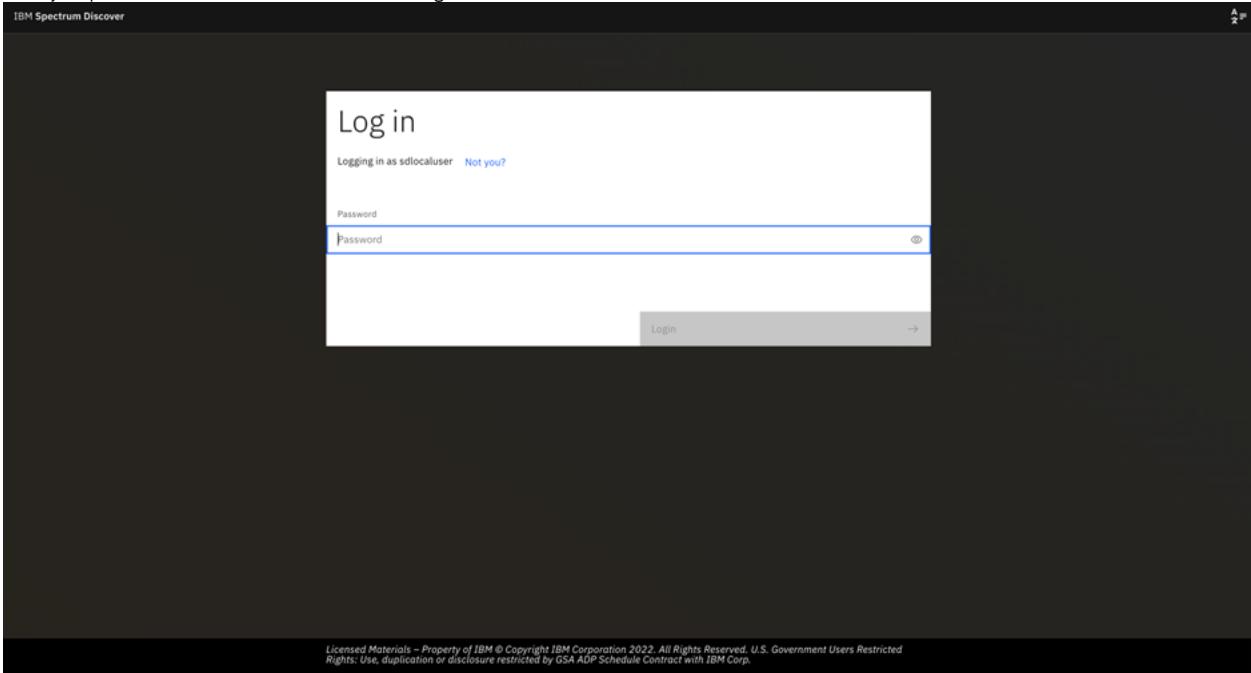
Log in to IBM Spectrum Discover Dashboard with SSO disabled

If you see a previous user logged in, you need to click on the Not you? button to see the following screen.

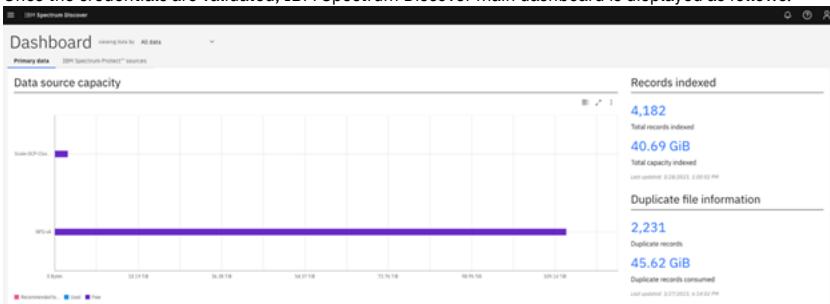
1. Enter your username in the User id field to access regular login and click Continue.



2. Enter your password in the Password field and click Login.



3. Once the credentials are validated, IBM Spectrum Discover 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 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 Spectrum Discover.
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 Spectrum Discover GUI.

When SSO is disabled as the default authentication method, the following default login credentials can be used for the IBM Spectrum Discover 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:

```
oc get pods -n ibm-data-cataloging | grep 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. Issue the following command to secure a connection to the IBM Spectrum® Discover application:

```
oc rsh -n ibm-data-cataloging spectrum-discover-keystone-599cf9fb77-bpqc8
```

Here, the default container is `spectrum-discover`.

3. Navigate to the directory `cd /` and run the `reset_sdadmin_details.sh` script to reset the details or the original password.

```
: ussur1
: 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/v1/users/<user_ID>/password: Post in 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 Spectrum® Discover. 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 Spectrum® Discover. 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 Spectrum Discover, 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 Spectrum Discover, the domain name is Default.

Description

Indicates the description of the user.

Figure 1. The Users tab

Username	Domain	Description
Ifrit	Default	
sdadmin	Default	
test_sd_0001	Default	
test_sd_0002	Default	
test_sd_0003	Default	

- [Creating user accounts](#)

The administrator can add local user accounts, which are authenticated by IBM Spectrum Discover, and assign roles to users.

Creating user accounts

The administrator can add local user accounts, which are authenticated by IBM Spectrum® Discover, and assign roles to users.

About this task

Navigate 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

- Log in to the IBM Spectrum Discover web interface.
- 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

Available Roles

Groups

- Enter a User Name and Email address for the user.
- Enter a Password for the user.
- 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 Spectrum® Discover. 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 Spectrum Discover, the domain name is Default.

Description

Indicates the description of the group.

Figure 1. The Groups tab

Name	Domain	Description
testgroup_sd_0001	Default	Description
testgroup_sd_0002	Default	Description
testgroup_sd_0003	Default	Description
testgroup_sd_0004	Default	Description

Items per page: 20 1–4 of 4 items 1 of 1 pages

- [Creating groups](#)

The administrator can add local groups that are authenticated by IBM Spectrum Discover and assign users and roles to the groups.

Creating groups

The administrator can add local groups that are authenticated by IBM Spectrum® Discover and assign users and roles to the groups.

About this task

Navigate 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

- 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

The screenshot shows the 'Add Local Group' page in the IBM Spectrum Discover interface. The 'Group Name' field is highlighted with a red border. The 'Available Roles' dropdown is open. The 'User' and 'Domain' sections are visible on the right. A note at the top states: 'The group name must be 5 to 20 characters and can only include "-" as a special character.' Another note below says: 'Assign Role to Group (Optional)'. The 'Description (Optional)' field is empty. At the bottom right are 'Cancel' and 'Save' buttons.

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 Spectrum® Discover with the collection.
- Assign a collection to a connection to associate specific metadata from that connection data source that is collected by IBM Spectrum Discover with the collection.

Users with the Data Admin role can view all metadata that is collected by IBM Spectrum Discover 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 Spectrum Discover, 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

The screenshot shows the 'Groups' section of the IBM Spectrum Discover interface. At the top, there are tabs for 'Groups', 'Users', 'Authentication domains', and 'Collections'. The 'Collections' tab is selected and highlighted with a blue border. Below the tabs is a search bar with a magnifying glass icon and a 'Create collection' button. A table lists one collection: 'spectrum-discover' with the description 'Bootstrap project for initializing the cloud.' At the bottom of the table are pagination controls showing '1 of 1 pages'.

- **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

The screenshot shows the 'Create Collection' window. It has two main sections: 'Name' and 'Members'. In the 'Name' section, there is a 'Collection name' input field containing 'spectrum-discover'. Below it is a 'Description (Optional)' field with the text 'Bootstrap project for initializing the cloud.'. There is also a checkbox labeled 'Create policy to tag files for this collection.' In the 'Members' section, there is a table with columns 'Name', 'Type', 'Domain', and 'Role'. A message at the top of the table says 'There are currently no users or groups in this collection.' Below the table is a blue 'Start to add members' button with a '+' sign. At the bottom right of the window are 'Cancel' and 'Create' buttons.

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 Spectrum® Discover 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.

- [Creating an LDAP connection](#)

The administrator can create a connection to a Lightweight Directory Access Protocol (LDAP) server that provides authentication for IBM Spectrum Discover 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 Spectrum® Discover users.

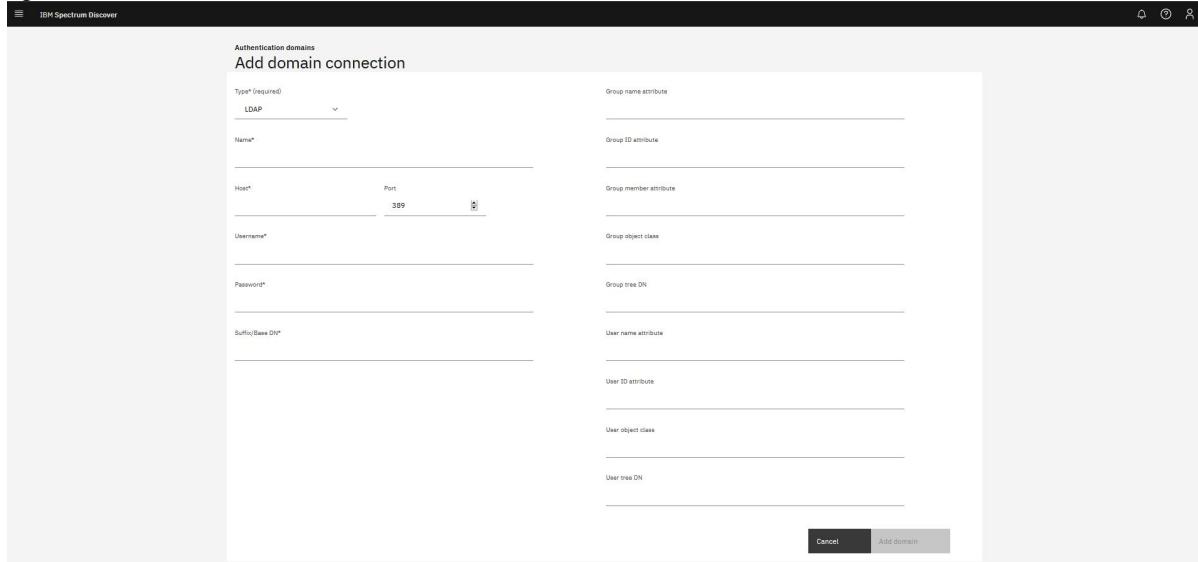
About this task

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.

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 Spectrum Discover 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 Spectrum® Discover 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 Spectrum Discover keystone pod trusted certificates list.

Procedure

1. Copy the self-signed certificate to the following directory, which is accessible to the keystone pod:

```
/opt/ibm/metaocean/data/keystone/cacerts/
```

If the cacerts directory is not available, you must create the directory.

2. Run the updated certificates script within the keystone pod by using the following command:

```
oc exec <spectrum-discover-keystone-pod-name> /update-cacerts.sh
```

3. Recycle the keystone pod by using the following command:

```
oc delete pod <spectrum-discover-keystone-pod-name>
```

Note: Change the certificate on IBM Spectrum Discover AD-server to 2048-bit. This secures the LDAPS domain connection.

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 Spectrum Discover. 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

The screenshot shows the 'Add domain connection' interface. At the top, it says 'Authentication domains' and 'Add domain connection'. Below that, there's a dropdown for 'Type*' which is set to 'IBM Cloud Object Storage'. There are four input fields: 'Name*' (empty), 'Host*' (empty), 'Port' (set to 389), and 'Username*' (empty). Below those are two more input fields: 'Password*' (empty) and 'Confirm password*' (empty). At the bottom right are two buttons: 'Cancel' and 'Add domain'.

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 Spectrum® Discover, 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 **c11** and data source **fs1**, the filter might be written as:

```
cluster='c11' 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:

- Enter the time of day to apply the policy by clicking the hour and minute from the widget that is shown.
- 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:

- Enter the time of day to apply the policy by clicking the hour and minute from the widget that is shown.
- 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 Spectrum Discover.

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 [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

The screenshot shows a table titled "Policies" with the following columns: Policy, Type, Schedule (UTC), Status, Progress, Collections, Last Modified by, and Last Modified. There is one row visible for "C3_start_stop". The "Type" column shows "DEEPMONITOR". The "Status" column shows "Active" with a "Paused" icon. The "Progress" column shows "30%" with "0 items out of 3000". The "Collections" column is empty. The "Last Modified by" and "Last Modified" columns show "sdadmin" and "2019-10-26T13:37:29.800Z" respectively. At the bottom left, there is a "Items per page" dropdown set to 20, and at the bottom right, a page number indicator "1 of 1 pages" followed by navigation arrows.

Policy	Type	Schedule (UTC)	Status	Progress	Collections	Last Modified by	Last Modified
C3_start_stop	DEEPMONITOR	Done	Active	30% 0 items out of 3000		sdadmin	2019-10-26T13:37:29.800Z

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

The screenshot shows a table titled "Policies" with the following columns: Policy, Type, Schedule (UTC), Status, Progress, Collections, Last Modified by, and Last Modified. There are two rows visible: "archivepol1" and "archivepol2". Both rows have "AUTOTAG" as the type and "Done" as the status. The progress is 100% for both. The last modified by user is "sdadmin" and the last modified date is "2020-01-21T12:13:31.000Z". The collections column is empty. At the bottom left, there is a "Items per page" dropdown set to 20, and at the bottom right, a page number indicator "1 of 1 pages" followed by navigation arrows.

Policy	Type	Schedule (UTC)	Status	Progress	Collections	Last Modified by	Last Modified
archivepol1	AUTOTAG	Done		100% 0 failed out of 71400		sdadmin	2020-01-21T12:13:31.000Z
archivepol2	AUTOTAG	Done		100% 0 failed out of 607		sdadmin	2020-01-21T12:38:12.000Z

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. <ul style="list-style-type: none"> • 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 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



The screenshot shows a table titled 'Policies' with a single row. The columns are: Policy, Type, Schedule (SFC), Status, Progress, Collections, Last Modified by, and Last Modified. The row contains: CS_test_001, DEEPINSPECT, None, active - stopped, 300%, 1 record out of 2000, system, 2019-10-28T17:37:29.80Z, and 2019-10-28T17:37:29.80Z. At the top of the table, there are buttons for Edit Policy, Delete Policy, Start, Stop Policy, Preview policy, and Add Policy. Below the table, there are pagination controls: Items per page: 20, 1 of 1 items, and a navigation bar with arrows and page numbers.

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: <ol style="list-style-type: none"> 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: <ol style="list-style-type: none"> 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 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:

podlog spectrum-discover-contentsearchagent

2. Any failures of download or inspection are logged in the application log file.

In the following example, the file fails during the inspection stage. The message shows that the failure is because the application cannot contact the Tika server.

```
[2019-04-23,16:45:54.945] agent[22665][ERROR][INSPECT]: Inspection failure-origpath  
metaocean1/alice.txt - error((HTTPConnectionPool(host='localhost',port=9998):Max retries  
exceeded  
with url:/tika(Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at  
0x7f4d813e0d0>:Failed to establish a new connection:[Errno 1111 Connection refused'.)))
```

```
oc logs $(oc get pods -n=spectrum-discover -l app=spectrum-discover-contentsearchagent -o=jsonpath={.items[0].metadata.name}) -n spectrum-discover
```

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 spectrum-discover scale --replicas=3  
    deploy/spectrum-discover-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 *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 Spectrum® Discover 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 pools](#) For more information, see *Internal Storage pools* in the *IBM Storage Scale: Administration Guide*.
- If you want to, you can configure IBM Storage Archive and its pools.
- Create the IBM Storage Scale connection in IBM Spectrum Discover. 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 Spectrum Discover ScaleILM application uses the same user, that is configured in IBM Spectrum Discover 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 *Data Cataloging: Concepts, Planning, and Deployment Guide*.

About this task

The IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover'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.

Make sure 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 *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 Spectrum Discover 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 (eadm). 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 Spectrum Discover 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:

- resndt
- 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 (resndt) 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:

```
podlog spectrum-discover-scaleilmdata mover
```

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:

```
podlog spectrum-discover-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 *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 Spectrum Discover 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 Spectrum Discover ScaleAFM application uses the same user that is configured in IBM Spectrum Discover 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 *Data Cataloging: Concepts, Planning, and Deployment Guide*.

About this task

The IBM Spectrum Discover 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 Spectrum Discover. The IBM Spectrum Discover 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 Spectrum Discover'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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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 filesset 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 filesset value in Spectrum scale_afm_filesset field.

It is configured as the AFM filesset 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 filesset. The field name must begin with a '/'. If not specified, the link path of the IBM Storage Scale AFM filesset 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 Spectrum Discover 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 Spectrum Discover 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

- Issue the following command to view the ScaleAFM application logs:

```
podlog spectrum-discover-scaleafmdata mover
```

- 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/lws3711.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'
```

- Run the following command to view policy engine pod logs:

```
podlog spectrum-discover-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_Poll]: Agent reported 'status: failed' for fkey: 'vault1lws3711.txt', path: 'vault1/lws3711.txt', reason: 'Failed queuing the file 'vault1/lws3711.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: 'vault1analytic results1.txt', path: 'vault1/analytics_results1.txt', reason: 'An exception occurred while processing message: Fileset 'DAAA_ICOS123' not found in file system 'fs1'', errno: 'EIO'
```

Exporting metadata to IBM Watson® Knowledge Catalog

IBM Spectrum® Discover 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 Spectrum Discover WKC connector waits for WKC credential values. In the absence of appropriate credentials, the Export button is not visible on the IBM Spectrum Discover 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 Spectrum Discover to Watson Knowledge Catalog .
- [Mapping similar source connections in Watson Knowledge Catalog](#)
IBM Spectrum Discover 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 Spectrum® Discover, 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 Spectrum Discover, 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover data source location. In this case, IBM Spectrum Discover 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 Spectrum® Discover 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 Spectrum Discover to a new or different Watson Knowledge Catalog connection.
- A new connection is added to IBM Spectrum Discover.

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 Spectrum® Discover instance and issue the following command to configure the WKC connector app:

```
oc -n spectrum-discover edit deploy/spectrum-discover-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 Spectrum Discover 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 scale deployment --replicas=0 spectrum-discover-wkcconnector  
oc scale deployment --replicas=1 spectrum-discover-wkcconnector
```

Exporting metadata

Export data with relevant metadata tags from IBM Spectrum® Discover 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover. 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 Spectrum Discover 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 Spectrum® Discover supports mapping source connections with Watson™ Knowledge Catalog connections (WKC) through WKC connector App.

When metadata is being exported to WKC, IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover but it is optional for WKC.

IBM Cloud Object Storage Infrastructure

For an IBM Spectrum Discover 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 Spectrum Discover. 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://modevmm19.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;modevmm19.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;modevmm19.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-
```

```
cb8edede242:local_cos_con,scale0;modevvm19.tuc.stqlabs.ibm.com:s3_scale,testbucket1.sd.ibm.com;s3.eu-west-1.amazonaws.com:s3_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 - WKConnector - ERROR - User authentication failed with response code 401
2020-07-16 12:46:04,871 - WKConnector - 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-wkconnector`

Incorrect WKC URL configuration

Resolve the errors that occur due to an incorrect WKC URL configuration.

The following errors occur:

```
HTTPSCConnectionPool(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 - WKConnector - 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-wkconnector`.

The following error occurs, when incorrect API key is provided for WKC cloud connections:

```
2020-09-28 13:29:30,189 - WKConnector - ERROR - Supplied API Key failed with response code 400
2020-09-28 13:29:30,190 - WKConnector - 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 - WKConnector - INFO - API Key not set, checking for user/password
2020-09-28 13:34:24,767 - WKConnector - ERROR - User authentication failed with response code 401
2020-09-28 13:34:24,768 - WKConnector - 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 - WKConnector - 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 Spectrum® Discover 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 Spectrum Discover.

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 - WKConnector - 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 - WKConnector - 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 WKConnector_work
```

Recycle Pod

```
oc scale deployment --replicas=1 spectrum-discover-wkcconnector
```

```
oc scale deployment --replicas=0 spectrum-discover-wkcconnector
```

If the Watson Knowledge Catalog connector pod remains down after you have performed the above steps, do the following:

1. When the Watson Knowledge Catalog connector (WKC connector) pod in IBM Spectrum® Discover cannot be started, find the pod name by using the following command:

```
kpl|grep spectrum-discover-wkcc
```

The sample output shows the pod detail similar to the following:

```
spectrum-discover spectrum-discover-wkcconnector-<unique_id> 1/1 Running 2d
```

2. Issue the following command to retrieve the logs from the WKC connector pod:

```
oc logs spectrum-discover-wkcconnector-<unique_id>
```

3. If the log terminates with an exception, check the policy engine log for errors by using the following command:

```
pollog|grep error
```

4. Check if you get the following errors at the time of the exception:

```
<time_stamp> - policy.policyapiservice - ERROR - b'Error: Cannot update application, 'WKConnector', it is in use by the following policies: [<policy_id>]. Please stop these policies'
```

5. The WKC connector pod does not start until the error shown in [step 4](#) is fixed. Therefore, the running policies must be stopped. Do the following to stop the running policies:

- a. Set up the environmental variables where **<sd_password>** is the IBM Spectrum Discover sdadmin password.

```
export SD_USER=sdadmin
export SD_PASSWORD=<sd_password>
export OVA=images
```

b. Issue the following command to retrieve the authorization token for curl:

```
gettoken
```

c. Issue the following command to retrieve the policies from IBM Spectrum Discover:

```
tcurl -H -k https://<spectrum_discover_url>:443/policyengine/v1/policies | jq
```

d. Issue the following command to retrieve the policy history from IBM Spectrum Discover:

```
tcurl -H -k https://<spectrum_discover_url>:443/policyengine/v1/policyhistory | jq
```

e. Issue the following command to retrieve individual policy history for a given <policy_id>:

```
tcurl -H -k https://<spectrum_discover_url>:443/policyengine/v1/policyhistory/<policy_id> | jq
```

f. Issue the following command to retrieve log of policy history for a given <history_id>:

```
tcurl -H -k https://<spectrum_discover_url>:443/policyengine/v1/policyhistory/<policy_id>/<history_id> | jq
```

6. For all the running policies, the <policy_id> for each must be terminated by using the following command:

```
tcurl -H -k https://<spectrum_discover_url>:443/policyengine/v1/policies/<policy_id>/kill -X POST | jq
```

7. Issue the following commands to recycle the pod:

```
oc scale deployment --replicas=1 spectrum-discover-wkcconnector
oc scale deployment --replicas=0 spectrum-discover-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 Spectrum® Discover 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 Spectrum Discover.
- The subsequent columns in the CSV file represent the tag values that can be imported into IBM Spectrum Discover for the associated object records.
- The second through Nth entries in the header row must correspond to valid tags in IBM Spectrum Discover 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,1,3,0,1,0,1
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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover.

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 Spectrum Discover, extending the records with new information.

Procedure

1. Configure a COS/S3 data source connection for IBM Spectrum Discover, with the vault or bucket that gets scanned later. The resulting system metadata that is indexed in IBM Spectrum Discover 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 *Data Cataloging: Concepts, Planning, and Deployment Guide*

2. Place external tag file on to the data source so that IBM Spectrum Discover 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 Spectrum Discover data source with connection name `camera_vault` is configured to scan vault and bucket `camera_lidar_semantic`.

3. Define tag names in IBM Spectrum Discover.

- a. Use the headers from the CSV file, starting with the second column.
- b. Create corresponding tags in IBM Spectrum Discover 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 *Data Cataloging: REST API Guide*.

4. Initiate the policy by using REST API interface. For more information, see [Initiating Policy Using REST API](#).

- [Initiating Policy Using REST API](#)

Initiating Policy Using REST API

- [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 Spectrum® Discover 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 Spectrum® Discover. To view the registered application, you can use the IBM Spectrum Discover GUI.

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_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'"  
}
```

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_connectio n	The Cloud Object Storage (COS) or S3 connection name that is defined in IBM Spectrum® Discover 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

- Send the following query request to list all the supported action IDs on IBM Spectrum Discover node:

```
curl -H "Authorization: Bearer <token>" -k https://<spectrum_discover_host>:443/policyengine/v1/action_ids
```

- Send the following query request to the IBM Spectrum Discover node to obtain the current Import Tags application schema:

```
curl -H "Authorization: Bearer <token>" -k  
https://<spectrum_discover_host>:443/policyengine/v1/applications/ImportTags/schema?action_id=IMPORT_TAGS
```

- 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 *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://<spectrum_discover_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 Spectrum® Discover 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 Spectrum Discover 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 *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 Spectrum Discover 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 'Authorization: Bearer <token>'  
https://<spectrum_discover_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, \"skipped_count\": 0, \"import_tags_count\": 0, \"run_id\": \"a3b0f76476443e8c5237aebe2c5408\"}",  
    "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:

```
podlog spectrum-discover-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 Spectrum® Discover 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 Spectrum Discover feature is leveraged 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover.
 - [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 Spectrum® Discover.

About this task

To enable the retention analytics for IBM Storage Protect archive data feature in IBM Spectrum Discover, you must first set an environment variable in IBM Spectrum Discover.

Following are the list of environment variable that you can set in the IBM Spectrum Discover. 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 Spectrum Discover server performs IBM Spectrum Discover 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 Spectrum Discover environment:

Procedure

1. Issue the following command to log in to the IBM Spectrum Discover by using ssh:

```
ssh <adminuser>@spectrumpdiscover.host
```

Note: The *<adminuser>* is a IBM Spectrum Discover server user who is assigned with an administrator role and the *spectrumpdiscover.host* is either the FDQN or IP of the IBM Spectrum Discover 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.

```

# 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
  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

```

- Save the preceding updated file and issue the following command to verify that the data is updated.

```
oc get configmap connmgr -o yaml
```

- Issue the following commands to verify that the update is made in the connection manager pod, the deployment must be scaled to 0 and then rescaled to 1:

```
oc scale -n spectrum-discover deployment/spectrum-discover-connmgr --replicas 0
```

Deployment is rescaled to 1.

```
oc scale -n spectrum-discover deployment/spectrum-discover-connmgr --replicas 1
```

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 Spectrum® Discover, 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

- Log in to the IBM Spectrum Discover GUI.
- Click  and go to Metadata > Tag Management
- On the Tags tab, click the Add Tag.
- 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.
- 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.

- 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.
- 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.
- Go to the Policies tab and click Add policy.
- On the Define page, enter the policy name in the Name field.
- Select `AUTOTAG` from Policy Type list.
- 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'.

- 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 Spectrum® Discover 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 Spectrum Discover 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 that is used to supplement storage system metadata with organization-specific information. For instance, an organization might segment their storage by project or by chargeback department. Those facets do not show up in the system metadata. Additionally, the storage systems themselves do not provide management and reporting capabilities based on those organizational concepts. Use custom tags to store additional information and manage, report, or search for data by using that organizationally important information.

Permissions

Security Administrators

Cannot create, update, delete or list any type of tag.

Data Administrators

Create, modify, delete, and list **Open**, **Restricted**, and **Characteristic** types of tags.

Data Users

Can list any type of tag, and can create and modify **Characteristic** tags.

Cannot create, modify or delete **Open** and **Restricted** tags.

Types of tags

Categorization

Categorization tags contain values such as project, department, and security classification. **Open** and **Restricted** type of tags are **Categorization** tags. Size limit is 256 bytes.

Characteristic

Characteristic tags can contain any value that is needed to describe or classify the object. Can contain lengthy values. Size limit is 4 KB.

Important: You cannot use group values when you search for characteristics. Use this tag specifically, for values that are not grouped.

- [Creating tags](#)

Use the following information when you create tags.

- [Viewing and searching tags](#)

- [Editing tags](#)

- [Deleting tags](#)

Creating tags

Use the following information when you create tags.

About this task

Use the Tags page to create new organizational tags. The table lists the tag name in the Field Name column, tag 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 New organizational Tag page, enter the name of the tag in the Name field.

Figure 1. New organizational tags

New organizational tags

Name

Type
▼

Values

 Press 'Enter' key to add the tag to the list

4. Select one of the following values from the Type menu:

Open

An Open tag can be anything that describes groups of records, but is non-restricted in value, such as project name, department, and sensor serial number.

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.

Characteristics

A Characteristics tag is something that is specific in value for each record. They are typically used for content extraction, such as patient name, VIN, or GPS location.

5. Enter one or more values for the tag into the Values box.

a. Press the Enter key to save each value. Each saved tag is displayed below the Values box.

6. Click the Submit button.

The tags, types, and values are displayed in the table in the Tags tab.

Viewing and searching tags

About this task

You can see a list of all tags or search for a subset of them on the Tags tab of the Metadata page.

Procedure

1. Go to **Metadata > Tag Management**.
2. Under Tags, a list of tag Names, tag Types, and tag Values appears.
3. Click the headings of each column to sort in ascending or descending alphabetical order.
4. Enter text into the Search box to find tags that begin with the text.
 As you enter text, a subset of the tags that contain the text string is automatically displayed.
5. Select a tag and then click **Edit** or **Delete** on the header row to perform the appropriate action.

Editing tags

About this task

You can edit tag values under the Tags tab of the Metadata page.

Procedure

1. Go to **Metadata > Tag Management**.
2. Select the tag and click the Edit pencil icon on the header row.
3. In the Edit organizational tags window that appears, enter one or more values for the tag in the Values box.
 Press **Enter** to save each value.
Note: The Name and the Type fields for the tag are unavailable for modifications.
4. Remove an existing tag value by clicking the "x" sign displayed next to the value bubbles.
5. Click **Submit**.
 The modified values are listed in the table under the Tags tab.

Deleting tags

About this task

You can delete tags on the Tags tab of 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 Spectrum® Discover. 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](#)
 - [Grouping data by file type](#)
Create policies to group data by file type.
 - [Searching system and custom metadata fields](#)

Searching

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 Spectrum® Discover 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 Spectrum® Discover 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

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 list below 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 Spectrum® Discover 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

resndnt

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 Spectrum Discover stores the S3 versioning identifier in the recordversion tag of the corresponding object. Cloud versioning allows the user 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 Spectrum® Discover 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 Spectrum Discover 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 "rxtnc" 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 Spectrum Discover 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:

The screenshot shows the IBM Spectrum Discover interface with the 'Query builder' tab selected. The 'SQL query' tab is active, displaying the following query:

```
Select * from Spectrum Discover where...
aces.entrytype like '%ACL'
```

A modal window titled 'Search query failed' is displayed, containing the error message: 'Invalid SQL input, please review syntax and try again.' Below the message, it says: 'Error: 400 Bad Request: Invalid group_by value. Cannot group_by with an ACL search.'

At the bottom of the query builder, there are filters for 'Group by' (set to 'Groups selected'), 'Limit results' (set to 10000), and 'Sort by' (set to 'e.g. owner asc').

Note: ACL grouping is not supported and in order 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 screenshot shows the 'Tag management' interface with the 'Tags' tab selected. A success message 'Tag created Tag Permission added' is displayed in a modal window.

The main table displays two rows:

Tag name	Type	Value
TEMPERATURE	Open	
Permission	Restricted	Read-Write, Read-Execute, Read-Only

At the top right of the interface, there are buttons for 'Add Tag' and a '+' sign.

The following policies are intended to set the permission tag based on the permissions flag:

Tag management

Tags Policies Policy history

Policy	Type	Schedule (UTC)	State	Status	Collections	Last modified by	Last modified
Accept_Read_Execute_Permissions	AUTOTAG	Done	<input type="radio"/> Inactive		spectrum-discover	sdadmin	12/8/2022, 8:29:50 PM
Accept_Read_Only_Permissions	AUTOTAG	Done	<input type="radio"/> Inactive		spectrum-discover	sdadmin	12/8/2022, 8:30:50 PM
Accept_Read_Write_Permissions	AUTOTAG	Done	<input type="radio"/> Inactive		spectrum-discover	sdadmin	12/8/2022, 8:29:01 PM

✓ Policy added

Accept_Read_Only_Permissions has been added

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"):

[← Back to query builder](#)

Query Results

Results view: [?](#)

Condensed **Individual**

Inspecting records **8,142**

Grouped By

- Access
- Cluster
- Datasource
- Filegroup
- Filename

[Show more...](#)

Matched 8,142 records from metaocean_view table in 0.718405 seconds

<input type="checkbox"/>	path	filename	↑ filetype	datasource	permission	aces permissions	aces flags
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Only	rwtTnNcCy	NULL
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Only	rwadxtTnNcCoy	g
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Only	rtnCyc	g
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Only	rwadxtTnNcCoy	NULL
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Only	rtnCyc	NULL
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Only	rtnCyc	NULL
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.jpg	jpg	Isilon-NFSv4Test	Read-Only	rwadxtTnNcCoy	g

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:

[← Back to query builder](#)

Query Results

Results view: [?](#)

Condensed **Individual**

Inspecting records **8,142**

Grouped By

- Access
- Cluster
- Datasource
- Filegroup
- Filename

[Show more...](#)

Matched 8,142 records from metaocean_view table in 0.615123 seconds

<input type="checkbox"/>	path	filename	↑ filetype	datasource	permission	aces permissions	aces flags
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Execute	rwadxtTnNcCoy	g
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Execute	rwtTnNcCy	NULL
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Execute	rtnCyc	g
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Execute	rtnCyc	NULL
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Execute	rwadxtTnNcCoy	NULL
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.docx	docx	Isilon-NFSv4Test	Read-Execute	rtnCyc	NULL
<input type="checkbox"/>	/ifs/sdiscover/Data/nfs4test/Nilesh /mail_files/	2mails_mail.jpg	jpg	Isilon-NFSv4Test	Read-Execute	rwadxtTnNcCoy	g

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 Spectrum® Discover 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 sanitization, and more. Data is identified by IBM Spectrum Discover 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 Spectrum Discover, which might or might not include enriched metadata for the records. If it does include enriched metadata, IBM Spectrum Discover 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

Application	Parameters	Action ID
ImportTags		"IMPORT_TAGS"
ScaleAFM		"COPY"
ScaleILM		"TIER"
contentsearchagent		"CONTENTSEARCH"
AFMDatamover		"COPY", "MOVE"

Items per page: 20 | 1–5 of 5 items | 1 of 1 pages

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 the *Application Registration REST API Guide*.

Using the IBM Spectrum Discover application catalog

Use the IBM Spectrum® Discover application catalog to search, download, or deploy applications (which are provided by IBM®, customers, or third parties) to use in IBM Spectrum Discover.

To use the commands in the examples throughout this document, you must use Secure Shell (SSH) to log in to the IBM Spectrum Discover. You also must have an authentication token that is generated from the command-line interface (CLI). The token expires after one hour. Run the following command to generate a token:

```
ssh moadmin@<your IP entered during mmconfigappliance>
# Enter in the password you set during the mmconfigappliance
export SD_USER=<sdadmin or another user with dataadmin privileges>
export SD_PASSWORD=<password for SD_USER above>
export OVA=images
gettoken
```

Note: In this example, **gettoken** is an alias under the **moadmin** user. Using an alias saves the token in an environment variable that is called **TOKEN**.

Note: The examples in the sections throughout this document use the aliases **tcurl** and **tcurl_json** under the **moadmin** user, which also uses the **TOKEN** environment variable.

Information about the endpoints

Follow the procedure to access information on endpoints:

1. Go to [IBM Spectrum Discover Documentation](#).
2. Choose the version of IBM Spectrum Discover that you are running.
3. Go to Table of Contents > REST API > Application management using APIs.

Querying the available applications

Run this command to query the applications that are available on **dockerhub**:

```
tcurl https://$(OVA)/api/application/appcatalog/publicregistry | jq
```

The output that is generated contains information that is gathered from the image itself (and from **dockerhub**).

Running an application as a Kubernetes pod

After you decide which application you want to run, from the query output, you can use it as a Kubernetes pod within IBM Spectrum Discover. Create a JSON-formatted file with the following information (the file that is created is named **example.json**):

```
{
  "repo_name": "ibmcom/spectrum-discover-example-application",
  "version": "1.2.3",
  "description": "Unique description about your use of this application",
  "application_name": "example",
  "my_env_var": "my_value",
  "LOG_LEVEL": "DEBUG"
}
```

Note: The attributes in the example can be explained as shown:

- The **repo_name** is the same **repo_name** that you used to download the application image.
- The **version** is the same as the version from the output of the **publicregistry** command.

- The **description** is a unique description that is based on your application use.
- The **application_name** is the name that gets registered within the **policyengine**. The system automatically appends a **-application** to the end of the file name for identification.

Run the following command to start the application as a Kubernetes pod:

```
tcurl_json https://localhost/api/application/appcatalog/helm -d@example.json -X POST | jq
```

You can add environment variables to the JSON example. These environment variables can be ones that your application needs or they can be ones that can override some software development kit (SDK) values. The application SDK supports the following environment variables that can override default settings:

LOG_LEVEL - INFO (default), DEBUG

Specifies the log level for the application to run with.

MAX_POLL_INTERVAL - 86400000 (in milliseconds)(default - 1 day)

Specifies when the Kafka consumer becomes unresponsive. Set this value higher than the time it takes for the application to process up to 100 records before it sends the reply to IBM Spectrum Discover. The default allows approximately 15 minutes for each record.

PRESERVE_STAT_TIME - False (default), True

Specifies whether to preserve atime or mtime when you run the deep-inspection application. If the application processes records from Network File System (NFS), Server Message Block (SMB), or local IBM Storage Scale connections, the system preserves the exact atime or mtime (in nanoseconds).

If the application processes records from a remote IBM Storage Scale connection, the system preserves atime or mtime up to and including seconds (with no subsecond preservation). The connection user must also have write access to the files. If the connection user does not have write access to the files, the system skips restoration of the atime or mtime because of permission errors. If **DEBUG** is on, you can see the original atime or mtime in the logs, so you can potentially manually restore any that fail.

Running an application as a python file

To run an application as a python file, you need to download the source code from [IBM Spectrum Discover Application catalog repository](#), available in the IBM public github repository. You can also build your own application or code and run as a python file. For more information about building your own application, see *Building your application* section in the [ExampleApplication repository](#) in the IBM github repository.

After you download or build your application source code, perform the following steps to run an application as a python file:

1. Issue the following command to install the required python packages:

```
sudo python3 -m pip install -r requirements.txt
```

2. Issue the following command to install the required OS packages. If you have any NFS connections, you must install the required packages.

```
sudo yum install nfs-utils
```

Note: For each connection you have in the administration page in the UI, the IBM Spectrum Discover application SDK creates the following:

- A sftp connection for each IBM Storage Scale connection
- A local NFS mount for each NFS connection
- A boto3 client for each COS connection

3. Define environment variables as shown.

```
export SPECTRUM_DISCOVER_HOST=https://<spectrum_discover_host>
# The IP or Fully Qualified Domain Name of your IBM Spectrum Discover instance.
# Default: https://localhost

export APPLICATION_NAME=<application_name>
# A short but descriptive name of your application.
# EX: exif-header-extractor-application or cos-x-amz-meta-extractor-application
# Default: sd_sample_application

export APPLICATION_USER=<application_user>
# A dataadmin user. Ex: sdadmin

export APPLICATION_USER_PASSWORD=<application_user_password>
# The password of the above dataadmin user.

export KAFKA_DIR=<directory_to_save_certificates>
# Directory where the kafka certs will be saved.
# Default: ./kafka

export LOG_LEVEL=<ERROR WARNING INFO DEBUG>
# Default: INFO
```

4. Start the sample application, by issuing the following command:

```
sudo -E python3 ./ExampleApplication.py
```

Scaling an application

An application by design processes each of the records one at a time. You can scale the number of replicas the pod is running to process records in parallel. You can scale up to 10 replicas based on the number of partitions available for the Kafka topics. Create a JSON-formatted file with the following information (the file that is created is named replicas.json):

```
{
  "replicas": 10
}
```

Then, run the following command to scale the replicas:

```
tcurl_json https://localhost/api/application/appcatalog/helm/interesting-anaconda-example-application -d@replicas.json -X PATCH
```

Note: In this example, `interesting-anaconda-example-application` is the combination of `deployment_name` and `chart_name` from the `Running an application` section.

Stopping an application

Run the following command to stop an application (no matter how many replicas you scale):

```
curl -X DELETE https://localhost/api/application/appcatalog/helm/interesting-anaconda | jq
```

Note: In this example, `interesting-anaconda` is the `chart_name` when the application was started.

- [Creating your own applications to use in the Data Cataloging application catalog](#)

Use this information to create applications for the Data Cataloging application catalog.

Creating your own applications to use in the Data Cataloging application catalog

Use this information to create applications for the Data Cataloging application catalog.

The following reference locations provide the source materials to help you start creating applications for Data Cataloging application catalog:

https://github.com/IBM/Spectrum_Discover_App_Catalog

This link contains the source code of the IBM®-provided applications.

https://github.com/IBM/Spectrum_Discover_Application_SDK

This link contains the source code for the IBM Spectrum® Discover Application Software Development Kit (IBM Spectrum Discover Application SDK). The link also describes how to build a test image for use in creating your own applications.

https://github.com/IBM/Spectrum_Discover_Example_Application

This link contains the source code for the template application. Start here when you create your own applications.

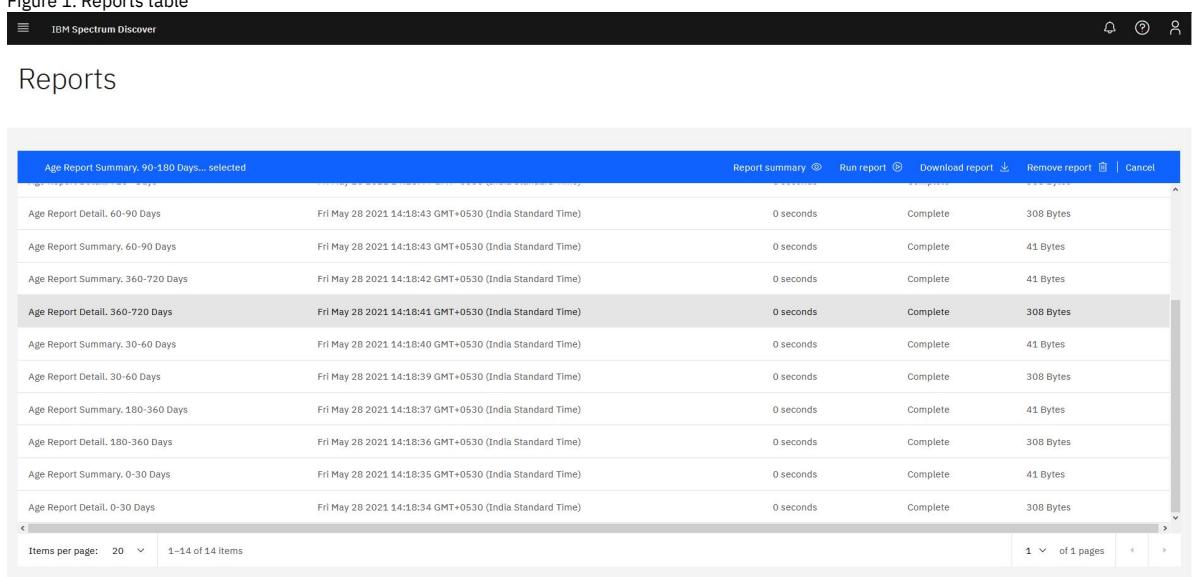
Reports

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 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



Age Report Summary: 90-180 Days... selected					Report summary	Run report	Download report	Remove report	Cancel
Age Report Detail. 60-90 Days	Fri May 28 2021 14:18:43 GMT+0530 (India Standard Time)	0 seconds	Complete	308 Bytes					
Age Report Summary. 60-90 Days	Fri May 28 2021 14:18:43 GMT+0530 (India Standard Time)	0 seconds	Complete	41 Bytes					
Age Report Summary. 360-720 Days	Fri May 28 2021 14:18:42 GMT+0530 (India Standard Time)	0 seconds	Complete	41 Bytes					
Age Report Detail. 360-720 Days	Fri May 28 2021 14:18:41 GMT+0530 (India Standard Time)	0 seconds	Complete	308 Bytes					
Age Report Summary. 30-60 Days	Fri May 28 2021 14:18:40 GMT+0530 (India Standard Time)	0 seconds	Complete	41 Bytes					
Age Report Detail. 30-60 Days	Fri May 28 2021 14:18:39 GMT+0530 (India Standard Time)	0 seconds	Complete	308 Bytes					
Age Report Summary. 180-360 Days	Fri May 28 2021 14:18:37 GMT+0530 (India Standard Time)	0 seconds	Complete	41 Bytes					
Age Report Detail. 180-360 Days	Fri May 28 2021 14:18:36 GMT+0530 (India Standard Time)	0 seconds	Complete	308 Bytes					
Age Report Summary. 0-30 Days	Fri May 28 2021 14:18:35 GMT+0530 (India Standard Time)	0 seconds	Complete	41 Bytes					
Age Report Detail. 0-30 Days	Fri May 28 2021 14:18:34 GMT+0530 (India Standard Time)	0 seconds	Complete	308 Bytes					

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

Status	Output size
Complete	41 Bytes
Complete	308 Bytes
Complete	41 Bytes
Complete	308 Bytes
Complete	308 Bytes
Complete	41 Bytes
Complete	41 Bytes
Complete	308 Bytes
Complete	41 Bytes

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

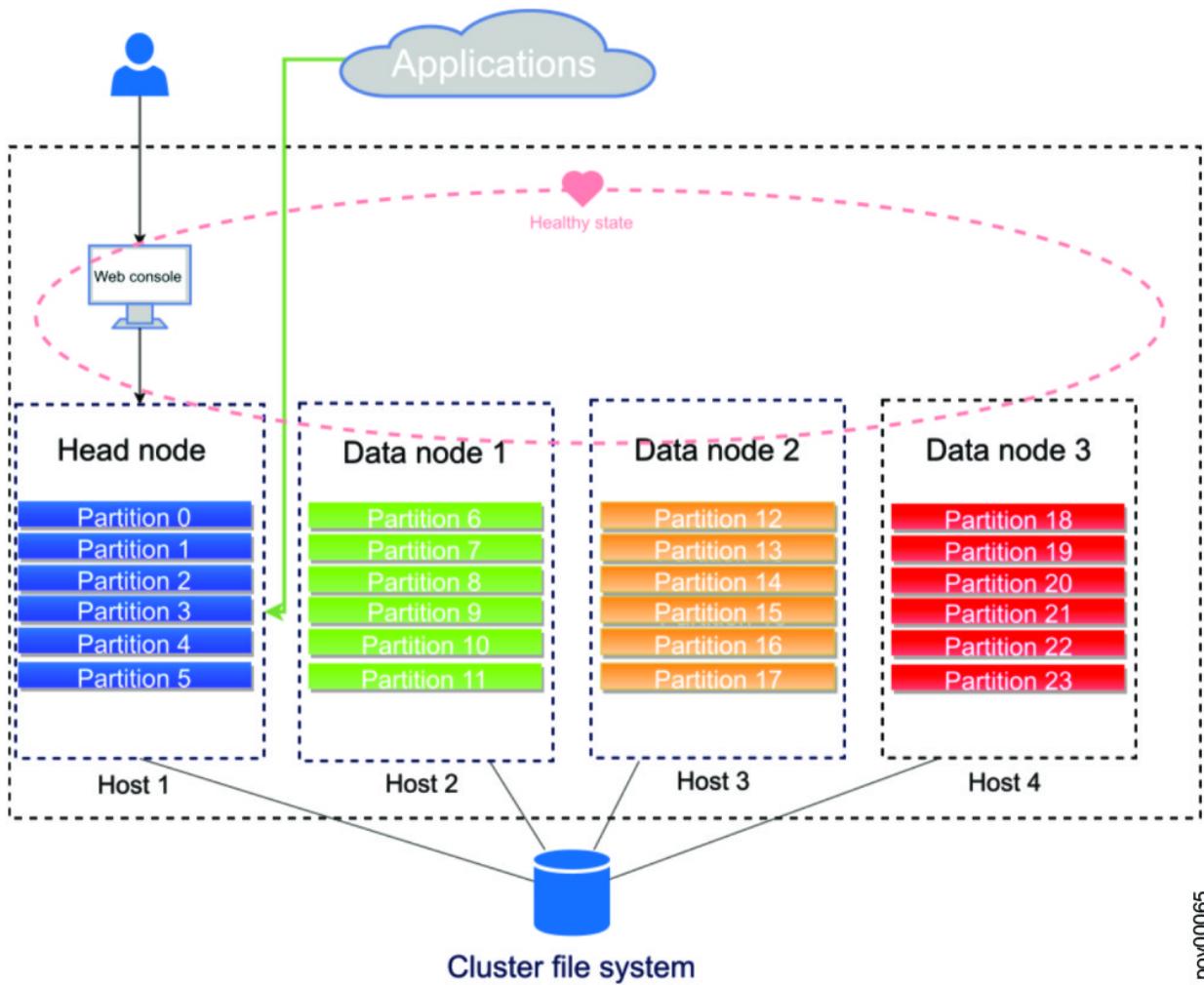
To remove a report, select a report and click Remove report.

High availability for a Db2 Warehouse MPP deployment

For an MPP deployment, Db2® Warehouse provides high availability, offering you the ability to have your data warehouse carry on with its activities if failures occur.

The HA solution is based on a heartbeat mechanism, automatic restart of services, and node failover. The heartbeat detects when a node, a database partition, or the web console is down, and the cluster manager takes the appropriate action. For instance, the cluster manager attempts to restart any failed data partitions or the web console. [Figure 1](#) shows a Db2 Warehouse HA group in a healthy state. The file system is not a part of the HA group, so use whatever HA solution that is appropriate for the technology you are using. Similarly, you can use a method such as a load balancer to make head node failures not apparent to connected applications.

Figure 1. Steady state for HA group

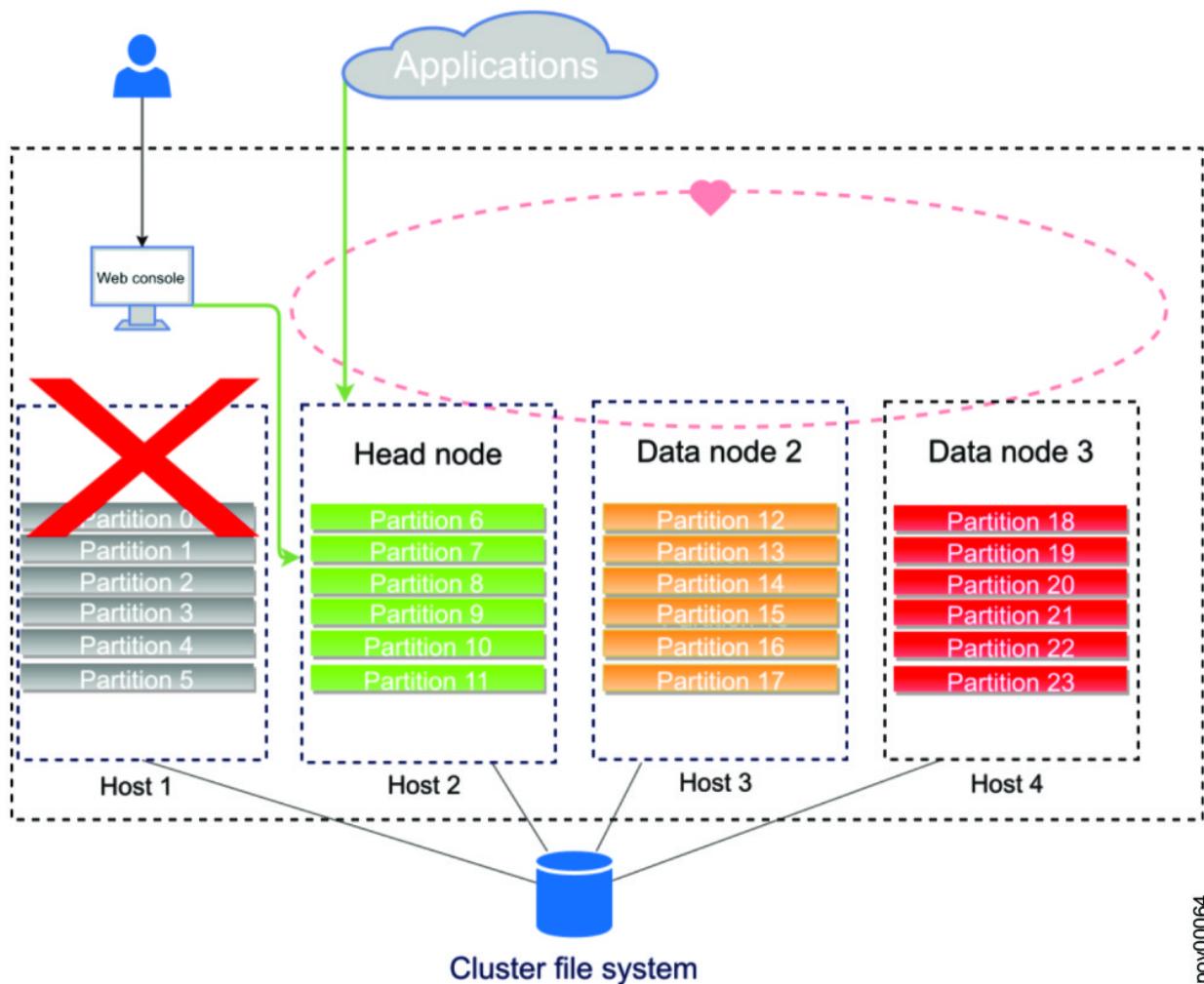


pov00065

If a data node fails and does not restart within the heartbeat interval, all services are stopped on that node. The data partitions (and their workload) that are assigned to that node are automatically redistributed across the surviving nodes in the cluster. There is no way to automatically reintegrate failed nodes; you must perform some manual steps to have a failed node rejoin the cluster.

If the head node fails and does not restart within the heartbeat interval, its data partitions are redistributed, and an election occurs. In the election, a new head node is selected from the first seven active data nodes in the cluster. As you can see in [Figure 2](#), the web console is restarted on the new head node.

Figure 2. HA group after head node failover



pov00064

After a head node failover, if the original head node becomes reachable again, restart the system for the original head node to become the current head node again.

- [Reintegrating a failed data node into an IBM Db2 Warehouse MPP cluster](#)
You must perform some manual steps to have a failed data node rejoin its cluster.

Reintegrating a failed data node into an IBM® Db2 Warehouse MPP cluster

You must perform some manual steps to have a failed data node rejoin its cluster.

About this task

To perform this task, you need to have root authority.

Procedure

1. Address whatever issue caused the node host failure.
2. Start the Db2® Warehouse container on the node you want to rejoin to the cluster.

```
docker start Db2wh
```

3. On the head node, stop the Db2 Warehouse services for the cluster.

```
docker exec -it Db2wh stop
```

4. On the head node, start the Db2 Warehouse services.

```
docker exec -it Db2wh start
```

The cluster should come up with the same topology as before the data node failure, with the data partitions distributed across all nodes.

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 Spectrum Discover environment.

- **[Recommended to move](#)**

In the IBM Spectrum Discover 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.

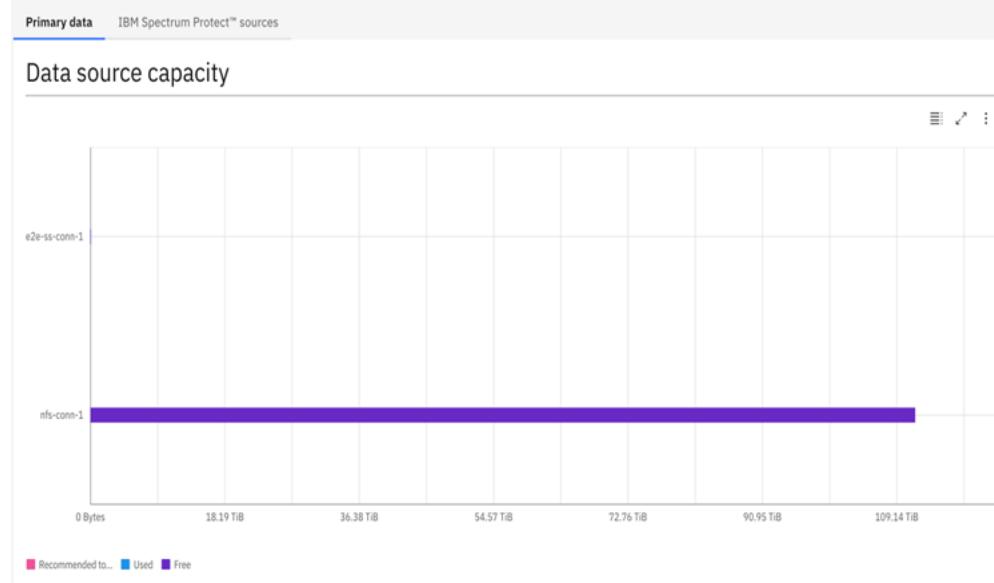
Viewing storage system capacity

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

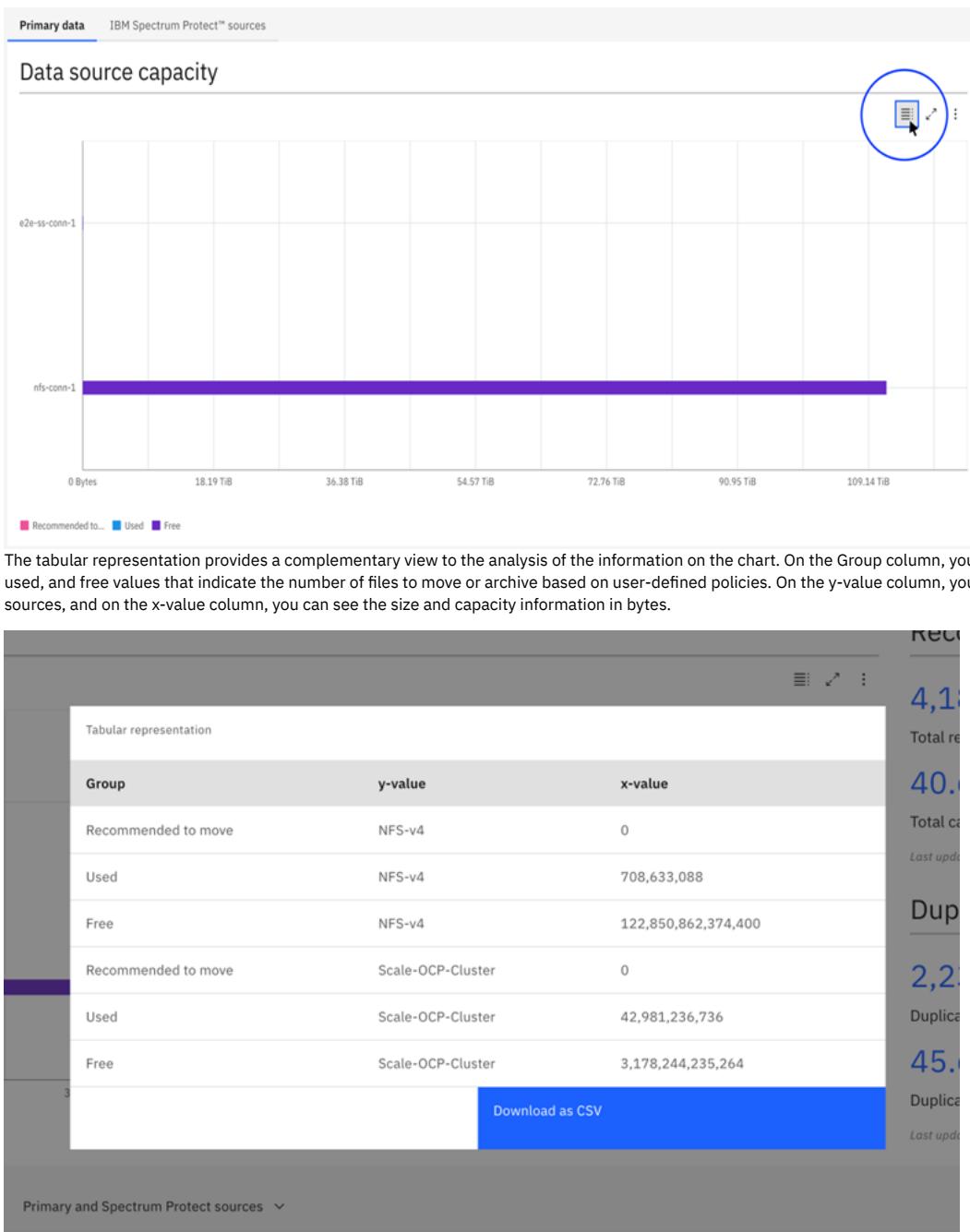
IBM Spectrum Discover collects size and capacity information. Generally:

- 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.



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.

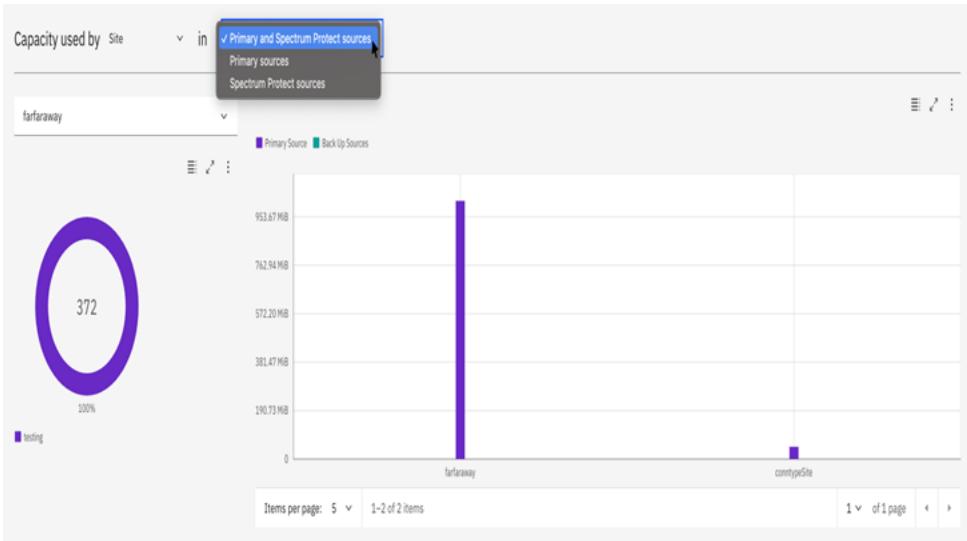
Use the Capacity Used by list to select an attribute and display the capacity consumers of that attribute in the graphs.

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.

Records Indexed

19,180,153

Total Records Indexed

322.41 TiB

Total Capacity Indexed

Last Updated : 2018-10-23 19:30:08

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.

Duplicate File Information

10,913,954

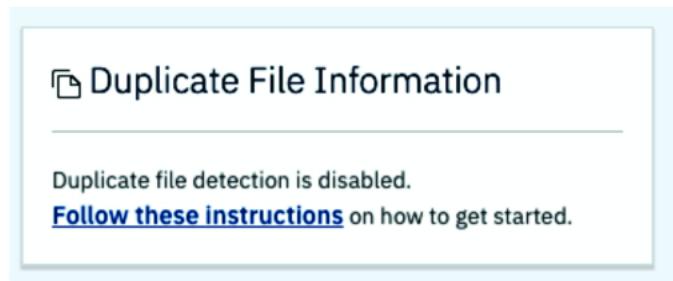
Duplicate Records

1.11 TiB

Total Capacity Consumed

Last Updated : 2018-10-23 00:15:39

Identifying potential duplicates can be resource-intensive on IBM Spectrum Discover. 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

Database Type	Auto-refresh	Last table refresh	Last refresh duration	Next scheduled refresh	Action
Metadata summarization database	On	1/6/2021, 4:00:02 pm	0:00:00	Every 30 minutes	Run table refresh
Duplicate record database	Off	31/5/2021, 8:40:55 pm	0:00:01		Run table refresh
Application catalog cache	On	1/6/2021, 3:30:04 pm	0:00:02	Every hour at :00	Run table refresh

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 Spectrum® Discover 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

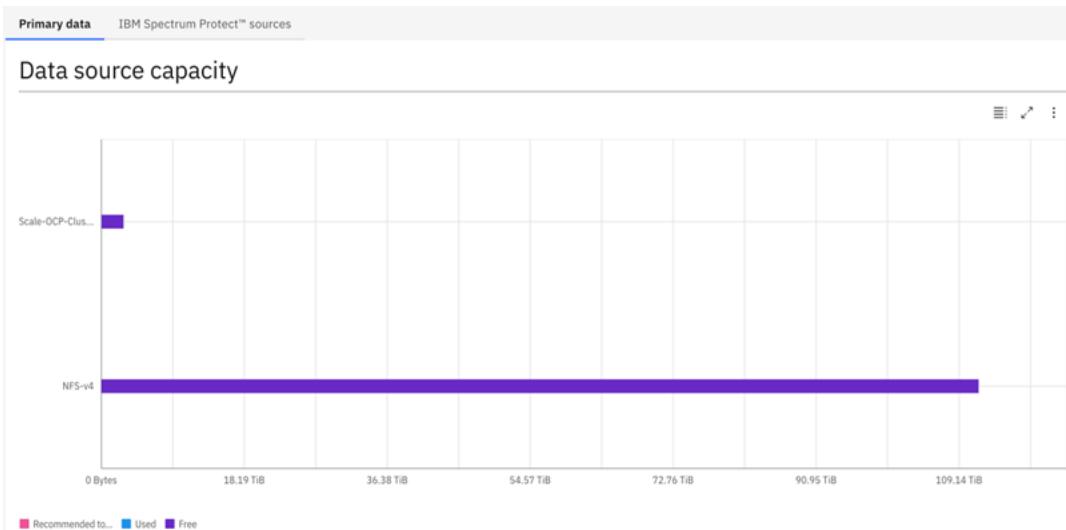
The physical location of the data source.

Recommended to move

In the IBM Spectrum® Discover 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 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.

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

Field Name	Type	Tags	Edit/Delete
COLLECTION	Open		
TEMPERATURE	Open		

[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

Policies

Modify a policy.

Policy type: AUTOTAG [?](#)

Inactive Active

Name

archive_pol

Filter

atime < (NOW() - 365 DAYS)

Extract tag from path

Tags

Field

Tag

TEMPERATURE

ARCHIVE



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](#).

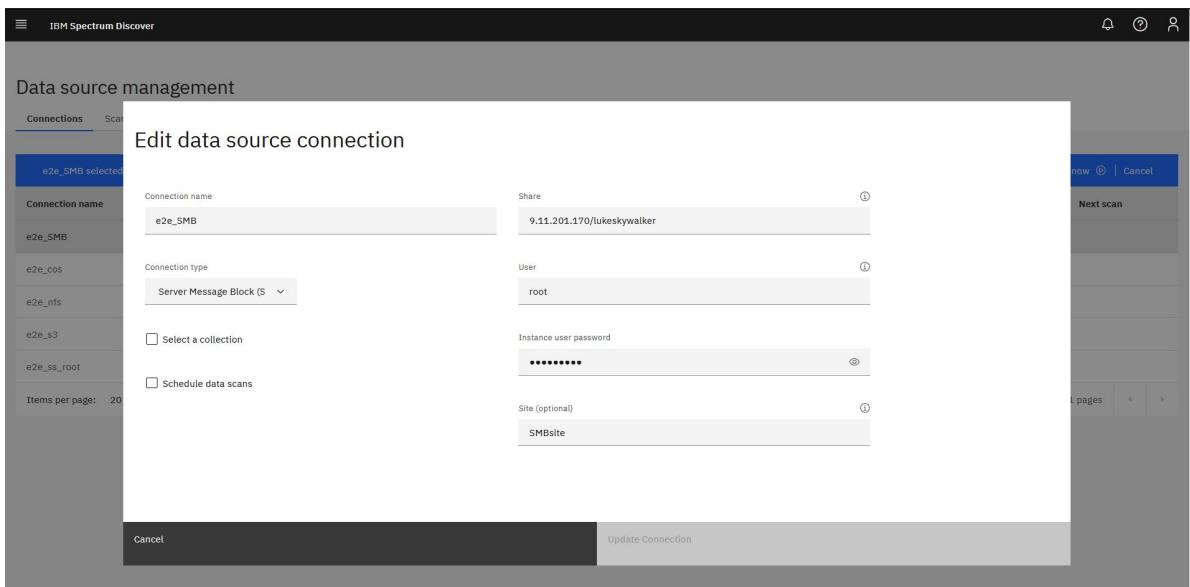
Figure 2. Starting the process to delete a data source connection

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 Data Cataloging service environment

You can monitor the health and status of the 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 Data Cataloging: REST API Guide*[REST API for 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 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.

Updating the network configuration

This topic describes how to update the network configuration.

Follow this procedure to update the network configuration of either master node or any of the worker nodes for IBM Spectrum® Discover. The process might take several hours because only one node can be completed at a time. For example:

1. Log in to the master node as the `moadmin` user.
2. Run the following command to change to the configuration directory:

```
cd /opt/ibm/metaocean/configuration
```

3. Run the following command to update your `old_fully_qualified_hostname`:

```
sudo ./mmconfigappliance -a <old_fully_qualified_hostname>
```

4. Update the network configuration of either the master or worker node to the new network configuration and make sure that the VM starts.

You can use the `sudo ./mmconfigappliance` command to update the network configuration. For example:

- a. Log in to node that is acquiring a new network configuration.
- b. Run the following command:

```
cd /opt/ibm/metaocean/configuration
```

c. Run the following command to change your configuration data:

```
sudo ./mmconfigappliance -n <new FQDN hostname>:<interface>:<new IP>:<netmask>:<gateway>:<dns>
```

5. Run the following command to update your `new_fully_qualified_hostname`:

```
sudo ./mmconfigappliance -b <new_fully_qualified_hostname>
```

You are prompted for the `moadmin` password.

The `old_fully_qualified_hostname` must be the old FQDN of either master or worker node that is to be updated. The `new_fully_qualified_hostname` must be the new FQDN of either master or worker node that is to be updated. Additionally, you must run both the `sudo ./mmconfigappliance -a <old_fully_qualified_hostname>` and `sudo ./mmconfigappliance -b <new_fully_qualified_hostname>` commands on the master node.

Using a third-party data movement application to manage data

Introduction to data movement with IBM Spectrum® Discover.

Before you begin

- The third-party application must be registered with IBM Spectrum Discover.
- IBM Spectrum Discover 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 Spectrum Discover can be extended by applications; for example, data mover applications can be used to move data between data sources, based on the IBM Spectrum Discover insights. Applications register with IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover can interact with third-party data movement applications to move, copy, or tier data between data sources. Create data management policies on IBM Spectrum Discover, 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 Spectrum Discover.

The third-party application registers with IBM Spectrum Discover 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 Spectrum Discover the user defines the filter, the parameter values, and when the policy must run. When the policy runs, IBM Spectrum Discover 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 Spectrum Discover. The summary is displayed to the user.

Note:

- During data migration, the migrated files need to preserve the IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover tracks the last known migration status for each file in the 'STATE' facet. This can be leveraged 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.

resndt (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, simply add "state = 'resndt'" to the filter criteria. To target files that are NOT resident, simply add "state <> 'resndt'" 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 Spectrum Discover and Moonwalk](#)

Policy-based data movement and management with integration of IBM Spectrum Discover and Moonwalk applications.

- [Preserving tags during data movement](#)

Data movement with IBM Spectrum Discover and Moonwalk

Policy-based data movement and management with integration of IBM Spectrum® Discover 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 Spectrum Discover, enabling to move, copy, or migrate/tier data from the IBM Spectrum Discover user interface.

The moonwalk application is certified against the IBM Spectrum Discover 2.0.4.

For more information about Moonwalk application and integrating the application with IBM Spectrum Discover, see [Moonwalk documentation](#).

Preserving tags during data movement

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
/	Thumbs.db	db	dir1	unknown	value1
/	Koala.jpg	jpg	dir1	jpg	value1
/	Desert.jpg	jpg	dir1	jpg	value1

About this task

If IBM Spectrum® Discover 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 Spectrum Discover.

Note:

You must perform a search using the above 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.

Enabling feature for skipping the snapshot directories

Enabling skipping feature allows you to skip the metadata ingestion to the snapshot directories.

Before you begin

- You must set the following environment variables in the IBM Spectrum® Discover:
 - `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 Spectrum Discover.

`NFS_SKIP_SNAPSHOT_DIRS`

This environment variable is used specifically for an NFS connection. When the variable is set to 'True', IBM Spectrum Discover 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 Spectrum Discover 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 commands to verify that the update is made in the connection manager pod. First, the deployment must be scaled to 0 and then rescaled to 1: Deployment is scaled to 0.

```
oc scale -n spectrum-discover deployment/spectrum-discover-connmgr --replicas 0
```

Deployment is rescaled to 1.

```
oc scale -n spectrum-discover deployment/spectrum-discover-conmgr --replicas 1
```

Data protection

- [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](#)

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 Data Cataloging environment. It must not be in use because it needs to scale 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 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 Data Cataloging, you need to scale pods manually.

Disaster recovery procedures

Use this process to recover from a disaster that involves an IBM Spectrum® Discover system and discusses the following scenarios:

- Recovery from the entire loss of a single node IBM Spectrum Discover deployment.
- Recovery from the entire loss of an IBM Spectrum Discover 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 Spectrum® Discover system as described in [Backup and restore](#).
2. Record the installation configuration, including:
 - Network settings
 - Storage settings
 - CPU and memory
 - IBM Spectrum Discover 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 Spectrum Discover. 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 Spectrum Discover version. For example, if IBM Spectrum Discover 2.0.0.3 is deployed and then upgraded to IBM Spectrum Discover 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](#).

REST API for Data Cataloging

- **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 Spectrum® Discover resource-based access control (RBAC) is a REST API service that enables role-based access to the IBM Spectrum Discover 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 Spectrum Discover.

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 Spectrum Discover 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, <spectrum_discover_host> is the host name or IP address of the API server:

- Endpoints for the Db2® Warehouse API:

```
https://<spectrum_discover_host>/db2whrest/v1/<endpoint_ID>
```

For example:

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_host>/db2whrest/v1/sql_query_async?<sql>
```

- Endpoints for the policy management API:

```
https://<spectrum_discover_host>/policyengine/v1/policies/<endpoint_ID>
```

For example:

```
curl -k -H 'Authorization: Bearer <token>' "https://<spectrum_discover_host>/policyengine/v1/policies/<policy_ID>"
```

- Endpoints for the resource-based access control (RBAC) API:

```
https://<spectrum_discover_host>/auth/v1/<endpoint_ID>
```

For example:

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_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://<spectrum_discover_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 Spectrum Discover APIs need to get an authentication token first by using their username and password. Then, use that token to get authenticated to the IBM Spectrum Discover system to perform various operations by using APIs.

Authentication is achieved in IBM Spectrum Discover 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 Spectrum Discover, the credentials are passed from IBM Spectrum Discover 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 Spectrum Discover integrates with both LDAP users and Cloud Storage Object users. Administrators can add the Cloud Storage Object domain to IBM Spectrum Discover and the users are imported into IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover services.

4. Users can access the IBM Spectrum Discover 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](#)
Changes 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

¹The 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://<spectrum_discover_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://<spectrum_discover_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": "\nbenjamin", "count": 2},
    {"owner": "\n_NULL_\n", "count": 989}, {"owner": "\nborgato", "count": 2},
    {"owner": "\nboston", "count": 4554}, {"owner": "\nroot", "count": 366104932},
    {"owner": "\nbeale", "count": 203545},
    {"owner": "\nbaldwin", "count": 2}, {"owner": "\nbehr", "count": 785144},
    {"owner": "\nbabcock", "count": 9082943}, {"owner": "\nbroadwood", "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": "\\\\'host\\", "count": 2,
  "sum": 16388}, {"filesystem": "\\\\'filesys1\\", "owner": "behr\\", "site": "\\\\'host\\",
  "count": 785144, "sum": 5000771899189}, {"filesystem": "\\\\'filesys1\\", "owner": "boston\\", "site": "\\\\'host\\",
  "count": 4554, "sum": 57670240959030}, {"filesystem": "\\\\'filesys1\\", "owner": "beale\\", "site": "\\\\'host\\", "count": 203545,
  "sum": 69505800364825}, {"filesystem": "\\\\'filesys1\\", "owner": "\root\\", "site": "\\\\'host\\",
  "count": 490, "sum": 265209686947}, {"filesystem": "\\\\'filesys1\\", "owner": "broadwood\\", "site": "\\\\'host\\",
  "count": 375, "sum": 48214210142151}, {"filesystem": "\\\\'filesys1\\", "owner": "benjamin\\", "site": "\\\\'host\\",
  "count": 2, "sum": 3140759161}, {"filesystem": "\\\\'filesys1\\", "owner": "babcock\\", "site": "\\\\'host\\",
  "count": 9082943, "sum": 48534270142857}, {"filesystem": "\\\\'filesys1\\", "owner": "borgato\\", "site": "\\\\'host\\",
  "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": "\\\\'host\\", "count": 2,
  {"filesystem": "\\\\'filesys1\\", "owner": "behr\\", "site": "\\\\'host\\", "count": 785144,
  {"filesystem": "\\\\'filesys1\\", "owner": "boston\\", "site": "\\\\'host\\", "count": 4554,
  {"filesystem": "\\\\'filesys1\\", "owner": "beale\\", "site": "\\\\'host\\", "count": 203545,
  {"filesystem": "\\\\'filesys1\\", "owner": "\root\\", "site": "\\\\'host\\", "count": 490,
  {"filesystem": "\\\\'filesys1\\", "owner": "broadwood\\", "site": "\\\\'host\\", "count": 375,
  {"filesystem": "\\\\'filesys1\\", "owner": "benjamin\\", "site": "\\\\'host\\", "count": 2,
  {"filesystem": "\\\\'filesys1\\", "owner": "babcock\\", "site": "\\\\'host\\", "count": 9082943,
  {"filesystem": "\\\\'filesys1\\", "owner": "borgato\\", "site": "\\\\'host\\", "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://<spectrum_discover_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_thym\\", "count": 54}, {"project": "\\\\'TCGA_prad\\", "count": 62}, {"project": "\\\\'TCGA_lam1\\", "count": 6}, {"project": "\\\\'TCGA_kirp\\", "count": 256}, {"project": "\\\\'hnsc\\", "count": 22}, {"project": "\\\\'TCGA_pcpg\\", "count": 2}, {"project": "\\\\'Eichler\\", "count": 877}, {"project": "\\\\'TCGA_uec\\", "count": 238}, {"project": "\\\\'TCGA_luad\\", "count": 1417}, {"project": "\\\\'Level1\\", "count": 6}, {"project": "\\\\'brca\\", "count": 14}, {"project": null, "count": 14}]}
]

```

```

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": 2, "sum": 2097216}, {"size": 3145824, "filename": "\asm_g149-C-ms22-cm0.mcdat", "count": 3, "sum": 3145824}, {"size": 4194432, "filename": "\asm_g151-C-ms22-cm0.mcdat", "count": 4, "sum": 4194432}, {"size": 2097216, "filename": "\asm_g153-C-ms22-cm0.mcdat", "count": 2, "sum": 2097216}, {"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": 2, "sum": 2097216}, {"size": 3145824, "filename": "\asm_g149-C-ms22-cm0.mcdat", "count": 3, "sum": 3145824}, {"size": 4194432, "filename": "\asm_g151-C-ms22-cm0.mcdat", "count": 4, "sum": 4194432}, {"size": 2097216, "filename": "\asm_g153-C-ms22-cm0.mcdat", "count": 2, "sum": 2097216}, {"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://<spectrum_discover_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": "\M01\", "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": \"fastal\", \"migstatus\": \"resndt\", \"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": "\M01\", "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\": \"fastal\", \"migstatus\": \"resndt\", \"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}], "count": 1
}

```

```

\"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\\": \"filesystem1\\",
\"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\\": \"filesystem1\\\", \"revision\\": \"M01\\\", \"site\\": \"\", ... \\\"owner\\": \"benjamin\\\", ... \\
{\\"filesystem\\": \"filesystem1\\\", \"revision\\": \"M01\\\", \"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://<spectrum_discover_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://<spectrum_discover_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 Spectrum® Discover 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": [{"datasource": "asc"}]
}

```

```

        "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"
        }
    ],
    "group_by": [],
    "sort_by": []
}
```

```

        "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 Spectrum Discover. Both exist on the IBM Spectrum Discover nodes:

- /opt/ibm/metaocean/reports/generate_report.py

To run the utilities, log in to an IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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://<spectrum_discover_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://<spectrum_discover_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

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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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 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://<spectrum_discover_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://<spectrum_discover_host>/db2whrest/v1/buckets -X GET -H "Accept: application/json"
```

- b. The following response is returned.

```
[{"name": "SizeRange", "type": "SizeRange", "ranges": [{"start": 0, "end": 1024}, {"start": 1024, "end": 2048}, {"start": 2048, "end": 4096}, {"start": 4096, "end": 8192}, {"start": 8192, "end": 16384}, {"start": 16384, "end": 32768}, {"start": 32768, "end": 65536}, {"start": 65536, "end": 131072}, {"start": 131072, "end": 262144}, {"start": 262144, "end": 524288}, {"start": 524288, "end": 1048576}, {"start": 1048576, "end": 2097152}, {"start": 2097152, "end": 4194304}, {"start": 4194304, "end": 8388608}, {"start": 8388608, "end": 16777216}, {"start": 16777216, "end": 33554432}, {"start": 33554432, "end": 67108864}, {"start": 67108864, "end": 134217728}, {"start": 134217728, "end": 268435456}, {"start": 268435456, "end": 536870912}, {"start": 536870912, "end": 1073741824}], "name": "TimeSinceAccess", "type": "TimeSinceAccess", "ranges": [{"start": 0, "end": 1000000000000000000}], "name": "FileGroup", "type": "FileGroup", "ranges": [{"start": 0, "end": 1000000000000000000}]}]
```

```
    }
```

```
1
```

/db2whrest/v1/buckets/<bucket>: GET

Gets information on a specific bucket that is supported.

Buckets influence the manner in which 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://<spectrum_discover_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://<spectrum_discover_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\"}  
  }  
]
```

/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://<spectrum_discover_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", "xslm", "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://<spectrum_discover_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://<spectrum_discover_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

¹The 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://<spectrum_discover_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://<spectrum_discover_host>/db2whrest/v1/sql_query?
select*20*from%20meteor%20limit%202 -X GET -H "Accept: application/json"
```

b. The following response is returned:

```
[{"datasource": "ctolib", "operation": "INDEX", "ingesttype": "SCAN", "revision": "M01", "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": "resnt", "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": "M01", "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": "resnt", "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://<spectrum_discover_host>/db2whrest/v1/sql_query?
select*20*from%20meteor%20limit%202 -X GET"
```

b. The following response is returned. The data is in CSV format:

```
0,ctolib,INDEX,SCAN,M01,,Spectrum Scale,ctolib.cluster1,250692,user1,group1,-rw-r--r--,brother,,,\ctolib/brother/dir1/,file1.txt,txt,,resnt,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,M01,,Spectrum Scale,ctolib.cluster1,250662,user1,group1,-rw-r--r--,brother,,,\ctolib/brother/dir1,file2..txt,txt,,resnt,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

¹The 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://<spectrum_discover_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.

- Write the record to be added to the database into a file named sql.dat.
- Submit the request:

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_host>/db2whrest/v1/sql_query -X POST -d"select * from datasrv04 where owner='root'"
```

- The following response is returned. The output is in plain text format:

```
0,filesystem,INDEX,SCAN,MO1,,Spectrum Scale,filesystem1.university.edu,3067343,root,root,  
-rw-r--r--,root,9,10,/filesystem1/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,  
sjdbList.out.tab,tab,resndt,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,filesystem1.university.edu/filesys13067343,  
/////////  
1,filesystem,INDEX,SCAN,MO1,,Spectrum Scale,filesystem1.university.edu,3067333,root,root,  
-rw-r--r--,root,9,10,/filesystem1/cellranger-2.0.0/refdata-cellranger-ercc92-1.2.0/star/,  
chrLength.txt,txt,resndt,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,filesystem1.university.edu/filesys13067333,  
/////////  
...
```

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,filesystem,INDEX,SCAN,MO1,,Spectrum Scale,filesystem1.university.edu,3067343,root,root,  
-rw-r--r--,root,9,10,/filesystem1/. . .  
1,filesystem,INDEX,SCAN,MO1,,Spectrum Scale,filesystem1.university.edu,3067333,root,root,  
-rw-r--r--,root,9,10,/filesystem1/. . .  
...
```

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.

- Write the SQL record in a file named sql.dat.
- Submit the request.

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_host>/db2whrest/v1/sql_query -X POST -d@"sql.dat" -H "accept: application/json"
```

- The following response is returned. The output is in JSON format:

```
[{"filesystem": "filesystem1", "operation": "INDEX", "source": "SCAN", "revision": "MO1", "site": "",  
"platform": "Spectrum Scale", "cluster": "filesystem1.university.edu", "inode": 3067112, "owner":  
"root", "group": "root", "permissions": "-r--r--r--", "fileset": "root", "uid": 9, "gid": 10, "path":  
"\\"/filesystem1\blast_db\\\"", "filename": "other_genomic.75.tar.gz", "filetype": "gz", "migstatus":  
"resndt", "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":  
"filesystem1.university.edu/filesys13067112", "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": "filesystem1", "operation": "INDEX", "source": "SCAN", "revision": "MO1", "site": "",  
"platform": "Spectrum Scale", "cluster": "filesystem1.university.edu", "inode": 3067092, "owner":  
"root", "group": "root", "permissions": "-r--r--r--", "fileset": "root", "uid": 9, "gid": 10,  
"path": "\\"/filesystem1\blast_db\\\"", "filename": "other_genomic.65.tar.gz", "filetype": "gz",  
"migstatus": "resndt", "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":  
"filesystem1.university.edu/filesys13067092", "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

¹The 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://<spectrum_discover_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.

- 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://<spectrum_discover_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://<spectrum_discover_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
```

- 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://<spectrum_discover_host>/db2whrest/v1/task_status/25b2df40-8f0c-4e32-a2f9-999ff3b18ada
```

- 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,resndt,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,resndt,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

¹The 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_host>/db2whrest/v1/task_status/3e40cfec-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://<spectrum_discover_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--", "filesset": "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--", "filesset": "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://<spectrum_discover_host>/db2whrest/v1/task_status/<task_id> -X GET
```

or

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_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://<spectrum_discover_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,M01,,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,resndt,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,M01,,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,resndt,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://<spectrum_discover_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 completed}
```

- 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://<spectrum_discover_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,M01,,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,resndt,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,M01,,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,resndt,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

¹The 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://<spectrum_discover_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://<spectrum_discover_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 tag 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 Data Cataloging system.

- [/policyengine/v1/regex: GET](#)

Retrieves list of all the regular expressions registered in the Data Cataloging system.

- [/policyengine/v1/regex/<regex_name>: GET](#)

Retrieves a single regular expression detail from the Data Cataloging system.

- [/policyengine/v1/regex: PUT](#)

Updates the regular expressions in the Data Cataloging system.

- [/policyengine/v1/regex: DELETE](#)

Deletes the regular expression with value from the 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://<spectrum_discover_host>/policyengine/v1/policies -X GET -H 'Accept: application/json'
```

or

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_host>/policyengine/v1/
policies/<policy_name>/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://<spectrum_discover_host>
/policyengine/v1/policies/policy1/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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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 <i>True</i> if the policy execution was for a user-defined policy or <i>False</i> 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
✓	✓	✓	✓	✗	✗

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_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://<spectrum discover 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 state 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, \"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\"}'}\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"
}

```

1

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
V	1V	1V	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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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 an 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://<spectrum_discover_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>`

'`<data>`' 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

¹The command allows access only if the schedule attribute is set to NOW.
²The 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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
✓	✓	✓	✗	✗	✗

Synopsis of the request URL

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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

¹READ/WRITE access to 'Open', 'Restricted', and 'Characteristics' types of tags.
²READ 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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 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 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\\.-]+\.\{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]\d{2}|222[1-9]|22[3-9]\d{2}|2[3-6]\d{2}|27[01]\d{2}|2720)\d{2}\b", "description": "Matching MasterCard number like: 5258704108753590"}, {"regex_id": "VisaCard", "pattern": "\b(4\d{3})\d{4}\b", "description": "Matching Visa Card numbers like: 4563-7568-5698-4587"}, {"regex_id": "AmexCard", "pattern": "\b3[47]\d{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|https|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", "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 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
✓	✓	✓	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 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 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://<spectrum_discover_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://<spectrum_discover_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",  
            "pattern": "(\\w+)/(\\w+)",  
            "replacement": "$2"  
        }  
    },  
    "schedule": "NOW",  
    "pol_filter": "user='research1'"  
}
```

```

        "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://<spectrum_discover_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.
- [`/connmgr/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://<spectrum_discover_host>/connmgr/v1/connections -X GET
```

or

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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 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	^✓	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://<spectrum_discover_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://<spectrum_discover_host>/connmgr/v1/scan/ssl -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://<spectrum_discover_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://<spectrum_discover_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 accesser system.

accesser_access_key

the access key for the corresponding accesser address specified by the **accesser_address**.

accesser_secret_key

The accesser'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 Spectrum® Discover 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 Spectrum Discover 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://<spectrum_discover_host>/connmgr/v1/connections -d '{"name": "con01", "platform": "Spectrum Scale", "cluster": "modevmm19.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://<spectrum_discover_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_info": {"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 <spectrum_discover_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

¹Collection 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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 above. 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://<spectrum_discover_host>/connmgr/v1/  
connections/con01 -d'{"name": "con02", "platform": "Spectrum Scale", "cluster":  
"modevmm19.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://<spectrum_discover_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://<spectrum_discover_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 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://<spectrum_discover_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:

- Obtain an auth token by using the credentials of the data admin user as shown:

```
curl -k https://<spectrum_discover_host>/auth/v1/token
-u "<user_name>:<password>"
```

For a valid user, the auth token is returned in the **X-Auth-Token** response header.

- Start the scan of a connection with name **sdconnection** using the following request:

```
curl -k -H "Authorization: Bearer <token>" -X POST
https://<spectrum_discover_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 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
V	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://<spectrum_discover_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:

- Obtain an auth token by using the credentials of the data admin user as shown:

```
curl -k https://<spectrum_discover_host>/auth/v1/token
-u "<user_name>:<password>"
```

For a valid user, the auth token is returned in the **X-Auth-Token** response header.

- Get the scan status of a connection with name **sdconnection** using the following request:

```
curl -k -H "Authorization: Bearer <token>" -X GET
https://<spectrum_discover_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 collections 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 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
V	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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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.	The stop action is not specified in the URL.
	Available action endpoints are: /pause, /resume, and /stop.	
	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 Spectrum® Discover uses. For more information, see [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 Spectrum Discover user with *data admin* role can register the application with IBM Spectrum Discover.

- [**/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.

- [**/api/application/appcatalog/publicregistry: GET**](#)

Queries the available docker images that match the unique query string from the hub.docker.com short description.

- [**/api/application/appcatalog/helm: GET**](#)

Returns a listing of all the application helm charts.

- [**/api/application/appcatalog/helm: POST**](#)

Creates and installs a helm chart (and it adds an `-application` suffix to the end of the name for easy retrieval).

- [**/api/application/appcatalog/helm: PATCH**](#)

Patches a helm chart to scale the deployment.

- [**/api/application/appcatalog/helm: DELETE**](#)

Deletes a helm chart.

/policyengine/v1/applications: POST

The /policyengine/v1/applications registers an action agent. Only an IBM Spectrum® Discover user with *data admin* role can register the application with IBM Spectrum Discover.

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://<spectrum_discover_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://<spectrum_discover_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 the valid 'action_id'
- The 'action_agent' parameter needs to be maximum 64 characters long and can contain alpha-numeric characters and ',', '_' or '-' characters.
- Action parameters, when converted to string, can be maximum 256 characters long.

3. Submit the following request:

```
curl -k https://<spectrum_discover_host>/policyengine/v1/applications -H "Content-Type: application/json" -X POST -d @agentregmsg.json -H "Authorization: Bearer <token>"
```

Note:

- Only 'deepinspect' is supported as the valid 'action_id'.
- The 'action_agent' parameter can be up to 64 characters long and contain alpha-numeric characters and ',', '_' or '-' characters.

Action parameters can be up to 256 characters long when converted to string.

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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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 Spectrum® Discover uses TLS protocol for encrypting the communication in-flight between the IBM Spectrum Discover 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 Spectrum Discover admin upon registration are used by applications to authenticate to the Kafka brokers in IBM Spectrum Discover.

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://<spectrum_discover_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://<spectrum_discover_host>/policyengine/v1/  
tlscert
```

Response:

```
-----BEGIN CERTIFICATE-----  
Samplej1CAmcCCQDqOCvwi/bLdzANBgkqhkiG9w0BQsFADCbMDELMAkGA1UEBhMCR0IxDjAMBgNV  
BAgMBUhBT1RTMRAwDgYDVQQHAdIdXJzbGV5MQwwCgYDVQQKDAhJk0xGTAXBgNVBAsMEHNwZWN0  
cnVtZGlzY292ZXIxGTAXBgNVBAMMEHnwZWN0cnVtZGlzY292ZXIxIzAhBgkqhkiG9w0BQEWFg1s  
YXdyZW5jZUB1ay5pYm0uY29tMB4XDTE4MTAxNTIyNTEzOVoxDTE5MTAxNTIyNTEzOVowbTELMAkG  
A1dyBhMCR0IxDjAMBgNVBAgTBUhbnRzMRAwDgYDVQQHewIdXJzbGV5MQwwCgYDVQQKewNJQk0x  
CzAJBgNVBAsTAK1EMSEw4wYDVQDEh0aG9yLnR1Yy5zGdsYWJzLmlibS5jb20wgEiMA0GCSqG  
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQCJFq8t8rk2fZf86TQWTE6R0VhFxmn9daqWyTQGz56zqGnX  
UxyF1wnIZQp7EGS3rBghEmV48X7gxxbshNvF1cr0jANvgElB66JOaESJu1m/s3B728qYHO4Wom  
2ii17hF3VdQhAvd72hd2kpJ3XVtp95yktLJ40Cr6x/4Kgsm8iKIiYYN3LbqqtfNB5CyKV9qNzTGb
```


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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_host>/policyengine/v1/actions -X GET -H 'Accept: application/json'
```

The following response is returned:

```
[ "WKCCConnector", "contentsearchagent", "ScaleILM" ]
```

/policyengine/v1/applications: DELETE

Deletes an application.

The following table shows which roles can access this REST API endpoint:


```

        "company_name": null,
        "company_url": null,
        "description": "",
        "filetypes": "",
        "icon_url": null,
        "maintainer": null,
        "parameters": null,
        "version": "",
        "license": "",
        "installed": 0,
        "installed_version": null,
        "instances": null,
        "additional_info": null,
        "update": "2019-11-12T18:00:03.802Z"
    },
    {
        "repo_name": "ibmcom/spectrum-discover-exif-header-extractor",
        "star_count": 0,
        "pull_count": 296,
        "created": "2019-11-08T20:56:24.160Z",
        "application_name": "exif_header_extractor",
        "company_name": null,
        "company_url": null,
        "description": "Extracts exif header information for jpeg, tiff files",
        "filetypes": "jpg,jpeg,tiff",
        "icon_url": null,
        "maintainer": null,
        "parameters": null,
        "version": "1.0.0",
        "license": "mit",
        "installed": 0,
        "installed_version": null,
        "instances": null,
        "additional_info": null,
        "update": "2019-11-12T18:00:03.822Z"
    }
]
}

```

The following example returns a specific image.

Request:

```
curl -X GET -k -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
https://<spectrum_discover_host>/api/application/appcatalog/publicregistry/ibmcom/spectrum-discover-exif-header-extractor | jq
```

Response:

```
{
    "success": "true",
    "message": "Successfully retrieved application: ibmcom/spectrum-discover-exif-header-extractor.",
    "output": {
        "repo_name": "ibmcom/spectrum-discover-exif-header-extractor",
        "star_count": 0,
        "pull_count": 296,
        "created": "2019-11-08T20:56:24.160Z",
        "application_name": "exif_header_extractor",
        "company_name": null,
        "company_url": null,
        "description": "Extracts exif header information for jpeg, tiff files",
        "filetypes": "jpg,jpeg,tiff",
        "icon_url": null,
        "maintainer": null,
        "parameters": null,
        "version": "1.0.0",
        "license": "mit",
        "installed": 0,
        "installed_version": null,
        "instances": null,
        "additional_info": null,
        "update": "2019-11-12T18:00:03.822Z"
    }
}
```

/api/application/appcatalog/helm: GET

Returns a listing of all the application helm charts.

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>" -H "Content-Type: application/json"
https://<spectrum_discover_host>/api/application/appcatalog/helm
```

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 all installed applications

Request:

```
curl -X GET -k -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"  
https://<spectrum_discover_host>/api/application/appcatalog/helm | jq
```

Response:

```
{  
    "success": "true",  
    "message": "Successfully retrieved installed applications.",  
    "output": [  
        {  
            "deployment": "exif-header-extractor-application-0-0-2",  
            "chart": "exif-header-extractor-application-0-0-2",  
            "repo": "ibmcom/spectrum-discover-exif-header-extractor",  
            "updated": "2020-09-25T01:02:49Z",  
            "replicas": 1  
        }  
    ]  
}
```

/api/application/appcatalog/helm: POST

Creates and installs a helm chart (and it adds an `-application` suffix to the end of the name for easy retrieval).

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>" -H "Content-Type: application/json"  
https://<spectrum_discover_host>/api/application/appcatalog/helm
```

Supported request types and response formats

Supported request types:

- POST

Supported response formats:

- JSON

Examples

The following example shows how to create an application as a helm chart. In the example:

```
application_name  
    The name of the chart.  
description  
    The description of the chart.  
version  
    The semver2 version number.  
repo_name  
    The name of the tagged image that is in the local docker registry.
```

For the contents of the exif.json file, enter this information:

```
[moadmin@spectrum-discover ~]$ cat exif.json  
{  
    "application_name": "exif-header-extractor",  
    "version": "0.0.2",  
    "description": "This extracts exif header information for jpg/jpeg or tiff files.",  
    "repo_name": "ibmcom/spectrum-discover-exif-header-extractor"  
}
```

Request:

```
curl -X POST -k -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"  
https://<spectrum_discover_host>/api/application/appcatalog/helm -d@exif.json | jq
```

Response:

```
{  
  "success": "true",  
  "message": "Successfully deployed application.",  
  "chart_name": "exif-header-extractor-application-0-0-2",  
  "deployment_name": "exif-header-extractor-application-0-0-2"  
}
```

/api/application/appcatalog/helm: PATCH

Patches a helm chart to scale the deployment.

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>" -H "Content-Type: application/json"  
https://<spectrum_discover_host>/api/application/appcatalog/helm
```

Supported request types and response formats

Supported request types:

- PATCH

Supported response formats:

- JSON

Examples

The following example shows how to patch a helm chart. In this example:

replicas

The number of replicas to scale the deployment to.

For the contents of replicas.json file, enter this information:

```
[moadmin@spectrum-discover ~]$ cat replicas.json  
{  
  "replicas": 5  
}
```

Note: The "exif-header-extractor-application-0-0-2" is the <deployment_name> from the helm POST response.

Request:

```
curl -X PATCH -k -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"  
https://<spectrum_discover_host>/api/application/appcatalog/helm/exif-header-extractor-application-0-0-2 -d@replicas.json | jq
```

Response:

```
{  
  "success": "true",  
  "message": "Successfully patched deployment.",  
  "output": {  
    "apiVersion": "ibm.spectrum.discover/v1alpha1",  
    "kind": "SpectrumDiscoverApplication",  
    "metadata": {  
      "creationTimestamp": "2020-09-25T16:32:28Z",  
      "generation": 2,  
      "name": "exif-header-extractor-application-0-0-2",  
      "namespace": "spectrum-discover",  
      "resourceVersion": "3985433",  
      "selfLink": "/apis/ibm.spectrum.discover/v1alpha1/namespaces/spectrum-discover/spectrumdiscoverapplications/exif-header-extractor-application-0-0-2",  
      "uid": "92986ff7-7b37-4fb5-a9c7-e08bfc5f448f"  
    },  
    "spec": {  
      "application_name": "exif-header-extractor-application-0-0-2",  
      "description": "This extracts exif header information for jpg/jpeg or tiff files.",  
      "env": {  
        "test1": "test2",  
        "test3": "test4"  
      },  
      "log_level": "INFO",  
      "replicas": 10,  
      "toleration": {  
        "key": "node-role.kubernetes.io/master",  
        "operator": "Exists",  
        "value": ""  
      }  
    }  
  }  
}
```

```

    "repo_name": "ibmcom/spectrum-discover-exif-header-extractor",
    "tag": "latest",
    "version": "0.0.2"
  },
  "status": {
    "conditions": [
      {
        "ansibleResult": {
          "changed": 0,
          "completion": "2020-09-25T16:32:48.519165",
          "failures": 0,
          "ok": 3,
          "skipped": 0
        },
        "lastTransitionTime": "2020-09-25T16:32:28Z",
        "message": "Awaiting next reconciliation",
        "reason": "Successful",
        "status": "True",
        "type": "Running"
      }
    ]
  }
}

```

/api/application/appcatalog/helm: DELETE

Deletes a helm chart.

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>" -H "Content-Type: application/json"
https://<spectrum_discover_host>/api/application/appcatalog/helm
```

Supported request types and response formats

Supported request types:

- DELETE

Supported response formats:

- JSON

Examples

The following example shows how to delete a helm file.

Request:

```
curl -X DELETE -k -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
https://<spectrum_discover_host>/api/application/appcatalog/helm/exif-header-extractor-application-0-0-2 | jq
```

Response:

```
{
  "success": "true",
  "message": "Successfully deleted Custom Resource: exif-header-extractor-application-0-0-2"
}
```

RBAC management by using APIs

The IBM Spectrum® Discover resource-based access control (RBAC) is a REST API service that enables role-based access to the IBM Spectrum Discover 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 Spectrum Discover.

The authentication service uses a default user with user name `sadmin` 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 Spectrum Discover 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 Spectrum Discover. The users and groups from the registered LDAP domain are automatically imported into IBM Spectrum Discover 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 Spectrum Discover 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 Spectrum Discover 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 Spectrum® Discover 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://<spectrum_discover_host>/auth/v1/token -v
curl -k -u cos/user1:pass https://<spectrum_discover_host>/auth/v1/token -v
curl -k -u cos/<access_key_id>:<secret_key> https://<spectrum_discover_host>/auth/v1/token -v
```

Without specifying domain name as part of user name:

```
curl -k -u user1:pass https://<spectrum_discover_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://<spectrum_discover_host>/auth/v1/token -v
```

Request that does not contain domain as part of the user name:

```
curl -k -u user1:pass https://<spectrum_discover_host>/auth/v1/token -v
```

Response:

```
< HTTP/1.0 200 OK
< X-Auth-Token: gAAAAABbUdgLOLk67Zk_XQ00bbJt_qe4suz449m7XLM-D6e0jUzDxBW574L38y5xF5y3wc-
Xq0Bp43uQH13QN-wTsioPoOfOlbyrN5avBag2iHUkCOgXyA46TvY5LdGu46LEs-gO-
qidQgUCg5im4QF3Zvw5enCTvQd2NlbAg186CVoc8Qk5o1ldJGd51nL3ELts0LsVg9F4oPhv9HixwZflhlCsuPg1Gw
< 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 Spectrum® Discover.

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_XQ00bbJt_qe4suz449m7XLM-
D6e0jUzDxBW574L38y5xF5y3wc-Xq0Bp43uQH13QN-wTsioPoOfOlbyrN5avBag2iHUkCOgXyA46TvY5LdGu46LEs-
gO-qidQgUCg5im4QF3Zvw5enCTvQd2NlbAg186CVoc8Qk5o1ldJGd51nL3ELts0LsVg9F4oPhv9HixwZflhlCsuPg1Gw'
https://<spectrum_discover_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

¹The 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://<spectrum_discover_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 users who are created in the system.

Request:

```
curl -k -H 'Authorization: Bearer <token>' https://<spectrum_discover_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": "a8dbddcd6b746b9958c34fd5a28855c",
      "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": "0337d1845a17fa29ff6ba83c2fb62870273030dfb4d10de8dfab2e9fbe68aef",
      "name": "user1",
      "options": {},
      "password_expires_at": null
    },
    {
      "domain_id": "bbd9f2993849402490cefc015013b6e9",
      "id": "d523a89af4e3c7042b564d241dc8058cd892030b63d00091128ba1d1d98a5d73",
      "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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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

¹The 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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

¹The 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://<spectrum_discover_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://<spectrum_discover_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

¹The 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://<spectrum_discover_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://<spectrum_discover_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

¹The 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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": "spectrum-discover",
          "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://<spectrum_discover_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://<spectrum_discover_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": "ldap",
      "id": "le8155daee6f021008a09aca09e1473d823ca7128d16ba25b3a10a7c1aa3433e",
      "name": "user2_NoGroup",
      "options": {},
      "password_expires_at": null
    },
    {
      "domain_id": "a69001e854174dc1a4e0c98f712744bb",
      "domain_name": "ldap",
      "id": "38f5434fe9d12fc3278deb29fb104441b87d040f5774bd545a2d1d0c7adaa7f",
      "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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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

¹The 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://<spectrum_discover_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://<spectrum_discover_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

¹The 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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 DELETE  
https://<spectrum_discover_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://<spectrum_discover_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
------------	-----------	------------------	-------	--------------

Data admin	Data user	Collection Admin	Admin	Service user
✓ ¹	✓ ¹	✓	✓	✓ ¹

¹The 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://<spectrum_discover_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://<spectrum_discover_host>/auth/v1/groups
```

Response:

```
(200 OK):
{
  "groups": [
    {
      "id": "a51f84bfeb034b12a9a362109d78bec3",
      "name": "test-group1",
      "description": "Test Group 1",
      "domain": "default"
    },
    {
      "id": "b62g95bfeb034b12a9a362109d89cf4",
      "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://<spectrum_discover_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://<spectrum_discover_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

¹The 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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": "09dd9136ea6416784d6c5bdf3fa2f78",  
          "is_domain": false,  
          "name": "spectrum-discover",  
          "parent_id": "default",  
          "tags": []  
        },  
        {  
          "description": "kutle",  
          "domain_id": "default",  
          "enabled": true,  
          "id": "40b0ea6fcfa7a4bafa38581cb508aa373",  
          "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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_host>/auth/v1/groups/a51f84bfeb034b12a9a362109d78bec3/user/4ee5b47f439640d29b6fac7253a64290 -X PUT  
-H "Content-Type: application/json"
```

Response:

200 if the command 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://<spectrum_discover_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://<spectrum_discover_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": "ldap",  
            "id": "d3b1ae0d1b5efb0c064797bf8239165cce32aaaf3c057151279198a780bc8cf0b",  
            "name": "Sales"  
        },  
        {  
            "domain_id": "a69001e854174dc1a4e0c98f712744bb",  
            "domain_name": "Sales",  
            "id": "d3b1ae0d1b5efb0c064797bf8239165cce32aaaf3c057151279198a780bc8cf0b",  
            "name": "Sales"  
        }  
    ]  
}
```

```

        "domain_name": "ldap",
        "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
✓	v ¹	v 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_host>/auth/v1/collections -d '{details of the collection}'
```

The details that can be specified are:

- Name
- Description
- Tags

Important: IBM Spectrum® Discover does not use collection tags. These collection tags are not related to the IBM Spectrum Discover 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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	✓	✓	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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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

Data admin	Data user	Collection Admin	Admin	Service user
✓	X	✓	✓	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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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.domainname ) 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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_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://<spectrum_discover_host>/auth/v1/domains/7a2251243611437cae707fb1aa5acf2c -X DELETE
```

Response:

```
204 No Content
```

Graceful shutdown

Provides detailed instructions to put Data Cataloging in idle state on an OpenShift® environment.

About this task

The procedure might be helpful in situations where 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 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 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 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 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, 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 Data Cataloging pods are expected.

- [Flushing Kafka topics](#)

You must have kafka to avoid issues related to restore the specified flushed out topic.

- [Returning Data Cataloging to a running state](#)

Provides detailed instructions to put 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 Data Cataloging to a running state

Provides detailed instructions to put Data Cataloging in running state on an OpenShift® environment.

About this task

Follow the steps in a specific sequence to put 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 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 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 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 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        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-6d6bbf8487-1jkws	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-6d6bbf8487-1jkws
```

Example output:

```
$oc delete pod isd-consumer-file-scan-6d6bbf8487-1jkws
pod "isd-consumer-file-scan-6d6bbf8487-1jkws" deleted
```

NAME	READY	STATUS	RESTARTS	AGE
isd-consumer-file-scan-6d6bbf8487-1jkws	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
```

Health check monitoring

Provides series of checkpoints to check Data Cataloging health status.

It provides 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 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-795544f666-125mc
$oc adm top pod isd-db2whrest-7c574c7bf-x719t
```

Example output:

```
$oc adm top pod isd-connmgr-795544f666-125mc
NAME           CPU (cores)   MEMORY (bytes)
isd-connmgr-scheduler-795544f666-125mc   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-795544f666-125mc|grep -A2 Limits
  Limits:
    cpu:      1
    memory:   4Gi
$oc describe pod isd-db2whrest-7c574c7bf-x719t|grep -A2 Limits
  Limits:
    cpu:      2
    memory:   8Gi
```

Note: 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 (milliseconds) 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-c719t      1/1     Running     0      43h
```

3. Get the logs and filter them.

For example:

```
$oc logs isd-db2whrest-7x574c7bf-c719t|grep -E '${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-6d6bbf8487-1jkw5	1/1	Running	0	7d13h
isd-consumer-file-scan-6d6bbf8487-m96nc	1/1	Running	0	7d13h
isd-consumer-file-scan-6d6bbf8487-w551r	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.

- 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
```

- 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 Data Cataloging API. For more information, see [REST API for 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.

- 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}
```

- 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 against 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
```

```

-----.
10364.

1 record(s) selected.

[db2inst1@c-isd-db2u-0 - Db2U ]$ select count(fkey) from bluadmin.metaocean
1
-----
10952.

1 record(s) selected.

```

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: spectrumsdiscover-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.

Collecting logs and metrics

Steps to collect logs and metrics for the 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
```

Creating a Data Cataloging application for metadata-based policies

Provides information about how to create Data Cataloging application for metadata-based policies.

By default, a 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 `nonroot` 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
```

Data Cataloging Harvester CLI

Data Cataloging Harvester is a new capability that is designed to import external data to the 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

- Make sure you have the following installations:
 - Python +3.11 or later version
 - curl tar gzip sed pip3 packages
 - Red Hat® OpenShift® CLI.

With the 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 Storage 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 Harvester 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 Data Cataloging harvester needs the following schemas. The 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 METAOCLEAN (
    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 Hat OpenShift CLI. For procedure, see [Red Hat OpenShift CLI](#).
2. 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

```

3. 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. Make sure 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

```

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 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-db2u0
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 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 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 Storage 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
```

Disaster recovery

Metro-DR is a disaster recovery solution supported by IBM Storage Fusion.

- Data Foundation can be configured in two OpenShift® clusters to use a shared stretched Ceph cluster. For information about steps to configure this solution, see [Configuring Data Foundation for Disaster Recovery](#).
- Regional-DR solution is designed to protect your applications against a wide range of large blast radius failures and disaster scenarios like data center failures. As an IBM Storage Fusion user, you can use Data Foundation to configure the Regional-DR solution that offers Disaster Recovery protection with ODF-based asynchronous volume replication for both block and file volumes. For more information about the solution, see [Regional-DR solution for Fusion Data Foundation](#).

Note: In disaster recovery, two clusters must be connected to failover and fallback applications. If the cluster recovers after the expiry of the client cert, clean and and setup the connection to rejoin the recovered cluster. For the procedure to rejoin, see [Reconnecting OpenShift Container Platform cluster](#).

Workloads

IBM Storage 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 Storage Fusion](#)
Mapping of IBM Cloud Paks with IBM Storage Fusion.

- [IBM Maximo® Applications support for IBM Storage Fusion](#)
Mapping of IBM Maximo® Applications with IBM Storage Fusion.

IBM Cloud Paks support for IBM Storage Fusion

Mapping of IBM Cloud Paks with IBM Storage Fusion.

IBM Cloud Paks	Version	Backup and restore support	Storage support
IBM Cloud Pak for Data Note: IBM Cloud Pak for Data supports backup and restore on both x86 and Power. IBM watsonx support in IBM Cloud Pak for Data: <ul style="list-style-type: none">• watsonx.data• watsonx.ai• watsonx.governance• watsonx Assistant	4.7.1 or later	<ul style="list-style-type: none"> • Online backup and recovery to the same or alternate cluster using IBM Storage Fusion • Backup and restore to the same cluster using IBM Cloud Pak for Data native tools 	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • IBM Cloud Pak for Data Storage considerations.
IBM Cloud Pak for Integration	2022.2 or later	Backup and restore using IBM Cloud Pak for Integration native tools	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • IBM Cloud Pak for Integration storage considerations
IBM Cloud Pak for Security	1.10 or later	Backup and restore using IBM Cloud Pak for Security native tools	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • IBM Cloud Pak for Security storage considerations
IBM Cloud Pak for Network Automation	2.4 or later	Backup and restore using IBM Cloud Pak for Network Automation native tools	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • IBM Cloud Pak for Network Automation storage considerations
IBM Cloud Pak for AIOps	4.5.0 or later	Backup and restore using IBM Cloud Pak for AIOps native tools <ul style="list-style-type: none"> • Online backup and recovery to the same or alternate cluster using IBM Storage Fusion (AIOps version 4.5 or later) • IBM Cloud Pak for AIOps backup and restore 	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • Storage considerations for IBM Cloud Pak for AIOps AI Manager
IBM Cloud Pak for Business Automation	21.0.3 or later	Backup and restore using IBM Cloud Pak for Business Automation native tools	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • IBM Cloud Pak for Business Automation storage considerations

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 Storage Fusion, the storage class must include `shared: true`.

For more information about the storage class, see [Storage provisioning using Container Storage Interface driver](#).

- [Validation tools for IBM Cloud Paks](#)

Run the storage readiness and performance suites on your IBM Storage 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 Storage Fusion environments. The suites run on OpenShift® Container Platform to evaluate the setup and performance of your storage setup.

Note: In Fusion Data Foundation and Red Hat® OpenShift Data Foundation environments, note that performance numbers may be lower than the published minimums due to the current use of `dsync` within the benchmark. It causes delays in Data Foundation environments where they honor the intent of having the write pass through cache and then be written to disk to be considered complete. In the completed benchmark testing using caching and write-to-disk scenarios (direct), Data Foundation can meet and surpass the published IBM Cloud Pak for Data benchmark requirements.

Storage validation tool

Run the IBM Cloud Pak for Data storage validation tool on your Red Hat OpenShift cluster to verify your storage setup for use with IBM Cloud Pak for Data. 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							Requirement
Cluster Name	PVC	Storage Type	Environment	Test Name	Thread Count	write MiB/s	
storage-performance-cluster	pvc-sysbench-rwo	Scale	HCI	rndwr_4k_8	8	46.4	Recommended to i
storage-performance-cluster	pvc-sysbench-rwx	Scale	HCI	seqwr_1g_2	2	3766.0	Recommended to i
storage-performance-cluster	pvc-sysbench-rwo	Scale	HCI	seqwr_1g_2	2	3901.2	Recommended to i
storage-performance-cluster	pvc-sysbench-rwx	Scale	HCI	rndwr_4k_8	8	46.1	Recommended to i

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 Storage Fusion HCI System configuration, that is, for a smallest environment. Depending on your configuration, environment, and network speed, the output would vary.

Detailed Measurements														
Cluster Name	PVC	Storage Type	Environment	Test Name	Thread Count	write MiB/s	Writes/s	read MiB/s	Reads/s	Total Time	Latency Min	Latency Avg	Latency Max	Latency 95th
storage-perf pvc-sysbenc1	Scale	HCI		rndwr_4k_8	8	46.4	11893.6	0	0	10	0.3	0.7	95.8	1
storage-perf pvc-sysbenc1	Scale	HCI		seqrd_1g_2	2	0	0	7380.8	7.2	10.2	217.2	279.6	583.2	555.4
storage-perf pvc-sysbenc1	Scale	HCI		rndrd_4k_8	8	0	0	3142.7	804539.1	10	0	0	2.1	0
storage-perf pvc-sysbenc1	Scale	HCI		seqwr_1g_2	2	3766	3.7	0	0	10.3	374.7	540	821.4	670.2
storage-perf pvc-sysbenc1	Scale	HCI		seqwr_1g_2	2	3901.2	3.8	0	0	10.4	354.6	521.3	737	695.1
storage-perf pvc-sysbenc1	Scale	HCI		rndwr_4k_8	8	46.1	11803.4	0	0	10	0.2	0.7	81.4	1.1
storage-perf pvc-sysbenc1	Scale	HCI		seqrd_1g_2	2	0	0	7404.1	7.2	10.2	207.3	280.6	585.3	548.3
storage-perf pvc-sysbenc1	Scale	HCI		rndrd_4k_8	8	0	0	3278.2	839227.1	10	0	0	37.7	0

Here, the cluster name is **storage-performance-cluster** of storage type **ocs**. The **Summary** section of the table also includes the requirement recommendations. In this example, for the **rndwr_4k_8** test, the write MiB/s is 46.5 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 Cloud Pak for Data sizing in the Sales Configurator

The starting point is a minimum IBM Cloud Pak for Data 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 Storage Fusion/storage sizing are also considered in your IBM Cloud Pak for Data sizing.

To view the sales configurator, go to <https://app.ibmsalesconfigurator.com/#/zen/configure>.

IBM Maximo® Applications support for IBM Storage Fusion

Mapping of IBM Maximo® Applications with IBM Storage Fusion.

Note: 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 Maximo Application Suite](#).

IBM Maximo® Applications	Version	Backup & Restore support	Storage support
Maximo Core	8.11.6 and later	Backup and restore to same or alternate cluster by following the IBM Maximo procedure	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • Red Hat® OpenShift® Data Foundation
Maximo Manage	8.7.4 and later	Backup and restore to same or alternate cluster by following the IBM Maximo procedure	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • Red Hat OpenShift Data Foundation
Maximo Health	Deployed as part of Manage	Backup and restore to same or alternate cluster by following the IBM Maximo procedure	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • Red Hat OpenShift Data Foundation

IBM Maximo® Applications	Version	Backup & Restore support	Storage support
Maximo IOT	8.8.4 and later	Backup and restore to same or alternate cluster by following the IBM Maximo procedure	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • Red Hat OpenShift Data Foundation
Maximo Monitor	8.11.3 and later	Backup and restore to same or alternate cluster by following the IBM Maximo procedure	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • Red Hat OpenShift Data Foundation
Maximo Predict	8.9.1 and later	Backup and restore to same or alternate cluster by following the IBM Maximo procedure	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • Red Hat OpenShift Data Foundation
Maximo Assist	8.8.1 and later	Backup and restore to same or alternate cluster by following the IBM Maximo procedure	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • Red Hat OpenShift Data Foundation
Maximo Visual Inspection Note: To check supported GPU models, see Maximo Visual Inspection documentation .	8.9.1 and later	Backup and restore to same or alternate cluster by following IBM Maximo Procedure	<ul style="list-style-type: none"> • IBM Storage Scale • IBM Fusion Data Foundation • Red Hat OpenShift Data Foundation

Knowing your IBM Storage Fusion user interface

Learn the various key elements of the IBM Storage Fusion user interface for ease of use and navigation.

To open the user interface from any of the supported browsers, do the following steps:

1. Log in to OpenShift® Container Platform using kubeadmin.
2. Click Applications icon in the title bar and click IBM Storage Fusion.
3. In the IBM Storage Fusion user interface, click Let's get started in the Welcome to IBM Storage Fusion page.

The Quick start page is the landing page of IBM Storage Fusion user interface.

From this Quick start page, you can provide storage for container workloads, backup and restore the applications and control plane data, investigate events and open support tickets, upgrade components, deploy hosts on IBM Cloud® Satellite.

The following options are available in the landing page of IBM Storage Fusion user interface:

- You can do the following tasks from the navigation menu:
 - Click Events to view IBM Storage Fusion events. They are Kubernetes native events that include filter-specific labels. For more information about the events page, see [Analyzing events in IBM Storage Fusion](#)
 - Click Applications to backup and restore application and its resources. For more information about how to backup and restore applications, see [Applications](#).
 - 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 Storage 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 Storage Fusion to navigate to IBM Storage Fusion.
 - The following details and options are available in the title bar:
 - Displays the cluster ID of IBM Storage 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 Storage Fusion.
 - Click About to know the version of IBM Storage Fusion.
 - The bell icon is visible on the IBM Storage 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 to 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 Storage Fusion.
- Note: To close all open sessions in IBM Storage Fusion, log out of both IBM Storage Fusion UI console and OpenShift Container Platform UI console.
- [Browser requirements](#)
List of supported web browsers for IBM Storage Fusion user interfaces.

- [Icons used in the user interface](#)

A reference to the icons used in the user interface and their naming convention.

Browser requirements

List of supported web browsers for IBM Storage Fusion user interfaces.

Browsers	Supported
Mozilla Firefox	Yes, it is supported. For more information about supported versions, see 2_versions
Google Chrome	Yes, it is supported. For more information about supported versions, see 2_versions
Safari	Yes, it is supported. For more information about supported versions, see 2_versions

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

Serviceability in IBM Storage Fusion

This section discusses the events and logs to monitor and troubleshoot issues.

Monitoring events

To understand Events page of IBM Storage Fusion, see [Analyzing events in IBM Storage Fusion](#).

Collecting logs

To create component-wise logs and download them for diagnosis, see [Collecting logs in IBM Storage Fusion](#).

- [Analyzing events in IBM Storage Fusion](#)

IBM Storage Fusion events are Kubernetes native events that include filter-specific labels.

- [Collecting logs in IBM Storage 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 Storage Fusion is used.

Analyzing events in IBM Storage Fusion

IBM Storage Fusion events are Kubernetes native events that include filter-specific labels.

The following component-based events and labels are available for IBM Storage Fusion:

Table 1. Component events and their labels

Component	Label
Installation component events	<code>isf.ibm.com/installer-component=spp</code>
Application events	<code>isf.ibm.com/application-name=<name of the application></code>
Assign policy to application events	<code>isf.ibm.com/application-name=<name of the application>, isf.ibm.com/backuppolicy-name=<Backup policy name></code>
Backup storage location events	
Backup events for daily backup	<code>isf.ibm.com/application-name=<name of the application>, isf.ibm.com/backuppolicy-name=<Backup policy name></code>
Restore application from backup	
Storage events for scale	<code>isf.ibm.com/storage-type=scale</code>

You can view these events from IBM Storage Fusion user interface, Red Hat® OpenShift® Container Platform console, or Kubernetes oc commands.

Steps to view events from IBM Storage 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 Storage 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 the events are not valid, then you can mark them as fixed. Run the API that is defined in the [Mark an event as fixed](#) to manually mark the event as fixed.

For a list of IBM Storage Fusion events and error messages, see [Events and error codes message references in IBM Storage 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.

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 Storage 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
```

Collecting logs in IBM Storage 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 Collect logs.

The Collect logs slide out pane gets displayed.

3. In the Collect logs slide out pane, select the logs of the following components. The components that are selected to be included in the log package.

Important:

- If a service is not enabled, the user interface does not display the service component in log collection.

Global Data Platform

Logs related to storage availability.

Data Foundation:

Logs related to K8 resources of openshift-storage.

If the Data Foundation service is configured to local mode then it includes the openshift-local-storage namespace. On HCP cluster, it includes the openshift-storage-client namespace.

Backup and restore

This option collects 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.

Note: For Hub and Spoke operations, collect logs on both the Hub and Spoke clusters.

Data Cataloging

This option collects the logs related to the Data Cataloging service. These logs are used to troubleshoot issues that are related to Data Cataloging. These logs contain mainly the **ibm-data-cataloging** namespace resources.

4. Click Collect.

The created log set gets added to the log list with status as Collecting. 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**

Components

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 log package to IBM via Call Home.
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

The browser upload option allows you to upload the log package via 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 Storage Fusion.

Log package

This section provides a quick overview of the log packages of the IBM Storage 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](#)

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 Storage Fusion namespace and the namespace where the service is deployed.

- `storage 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 Storage 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 Storage 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, deamon 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.

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 Storage 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, deamon 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.

Benefits of enabling call home

Configure and enable the call home feature as it improves the user experience when IBM Storage 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 Storage Fusion](#).

However, if the call home feature is not enabled, then the IBM support team first collects debugging data from the IBM Storage 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 Storage 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 Storage Fusion system:*

If the call home feature is enabled, then a customer can easily 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 Storage Fusion user interface.

- [**Data privacy with call home**](#)

A licensee can configure the IBM Storage Fusion to automatically send specific cluster information to the IBM® support team by using the Call Home feature.

Data privacy with call home

A licensee can configure the IBM Storage 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 Storage 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

Steps to enable and disable Call Home from the IBM Storage Fusion user interface. In addition, you can edit Call Home configuration details.

Before you begin

Ensure that the firewall sites must be configured for Call Home:

- esupport.ibm.com

- .secure.ecurep.ibm.com
- .docker.com

About this task

If you enable Call Home, the system automatically creates a ticket whenever a critical problem occurs. It helps the service representative to debug the issues.

Procedure

1. From the IBM Storage Fusion menu, click Settings > Call Home.
2. In the Call Home section, click Enable.
The Enable Call Home page gets displayed.
3. Enter Customer number and Contact email.
4. Click Save.
The Call Home gets enabled and the details are saved. You can use the Edit icon to update values of Customer number and Contact email.
5. Click Verify connection to test Call Home post enablement.
It also gets triggered automatically when Call Home is enabled.
6. To disable or update details, click the Edit icon.
 - Edit details
Make changes to the details and click Save. A success confirmation message gets displayed.
 - Disable Call Home
Click Disable to disable the Call Home. The Disable Call Home window opens and click Disable. If you want to save your details before disabling the Call Home, select Remember Call Home set-up information.

IBM Storage Fusion Data Foundation

- [**Release notes**](#)
The release notes for IBM Storage Fusion Data Foundation 4.15 summarizes all new features and enhancements, notable technical changes, and any known bugs upon general availability.
- [**Introduction to Fusion Data Foundation**](#)
IBM Storage Fusion Data Foundation is a highly integrated collection of cloud storage and data services for Red Hat OpenShift Container Platform. It is available as part of the Red Hat OpenShift Container Platform service catalog, packaged as an operator 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 Storage Fusion Data Foundation deployment.
- [**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 Storage Fusion Data Foundation.
- [**Managing hybrid and multicloud resource**](#)
Use this information to learn how to manage hybrid and multicloud resources.
- [**Replacing nodes**](#)
Safely replace a node in a Fusion Data Foundation cluster.
- [**Replacing devices**](#)
Safely replace storage devices for Fusion Data Foundation.
- [**Monitoring Fusion Data Foundation**](#)
Learn how to monitor IBM Storage Fusion Data Foundation using the Block and File, and Object dashboards.
- [**Troubleshooting**](#)
Administrators can use this troubleshooting information to understand how to troubleshoot and fix their IBM Storage Fusion Data Foundation cluster.
- [**Configuring Data Foundation for Disaster Recovery**](#)
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.

Release notes

The release notes for IBM Storage Fusion Data Foundation 4.15 summarizes all new features and enhancements, notable technical changes, and any known bugs upon general availability.

IBM Storage 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.

IBM Storage Fusion Data Foundation is integrated into the latest Red Hat OpenShift Container Platform to address platform services, application portability, and persistence challenges. It provides a highly scalable backend for the next generation of cloud-native applications, built on a technology stack that includes IBM Storage Ceph, the Rook Operator, and NooBaa's Multicloud Object Gateway technology.

IBM Storage Fusion Data Foundation provides a trusted, enterprise-grade application development environment that simplifies and enhances the user experience across the application lifecycle in a number of ways:

- Provides block storage for databases.
- Shared file storage for continuous integration, messaging, and data aggregation.
- Object storage for cloud-first development, archival, backup, and media storage.

- Scale applications and data exponentially.
 - Attach and detach persistent data volumes at an accelerated rate.
 - Stretch clusters across multiple data-centers or availability zones.
 - Establish a comprehensive application container registry.
 - Support the next generation of OpenShift workloads such as Data Analytics, Artificial Intelligence, Machine Learning, Deep Learning, and Internet of Things (IoT).
 - Dynamically provision not only application containers, but data service volumes and containers, as well as additional OpenShift Container Platform nodes, Elastic Block Store (EBS) volumes and other infrastructure services.
- **[About this release](#)**
IBM Storage Fusion Data Foundation 4.15 is now available. This release includes new enhancements, features, and known issues. IBM Storage Fusion Data Foundation 4.15 is supported on the Red Hat OpenShift Container Platform version 4.15.
- **[New features](#)**
This section describes new features introduced in IBM Storage Fusion Data Foundation 4.15.
- **[Enhancements](#)**
This section describes the major enhancements introduced in IBM Storage Fusion Data Foundation 4.15.
- **[Technology Previews](#)**
This section describes the technology preview features introduced in IBM Storage Fusion Data Foundation 4.15 under Technology Preview support limitations.
- **[Bug fixes](#)**
This section describes the notable bug fixes introduced in IBM Storage Fusion Data Foundation 4.15.
- **[Known issues](#)**
This section describes the known issues in IBM Storage Fusion Data Foundation 4.15.
-

About this release

IBM Storage Fusion Data Foundation 4.15 is now available. This release includes new enhancements, features, and known issues. IBM Storage Fusion Data Foundation 4.15 is supported on the Red Hat OpenShift Container Platform version 4.15.

IBM Storage Fusion Data Foundation is included in the following [Red Hat Product Errata](#):

- [RHSA-2024:1383](#)
-

New features

This section describes new features introduced in IBM Storage Fusion Data Foundation 4.15.

- [Support for multiple storage clusters](#)
- [Non resilient storage class](#)
- [Recovering to a replacement cluster for Metro-DR](#)
- [Red Hat OpenShift virtualization workloads for Metro-DR](#)
- [Support setting of RBD storage class as the default](#)
- [Performance profiles](#)
- [Ability to create backing stores in Red Hat OpenShift cluster that use AWS Security Token Service](#)
- [Runbooks for Fusion Data Foundation alerts](#)
- [Allow expansion of encrypted RBD volumes](#)
- [Improved cluster availability with additional monitor daemon components](#)
- [Alerts for monitoring system overload](#)
- [Shallow volumes support for snapshot or clone](#)

Support for multiple storage clusters

IBM Storage Fusion Data Foundation provides the ability to deploy two storage clusters, one in internal mode and the other in external mode. The first cluster must be installed in internal mode in the `openshift-storage` namespace and the second cluster in external mode in the `openshift-storage-extended` namespace. Vice-versa is currently not supported.

Non resilient storage class

Fusion Data Foundation allows the addition and use of a new non resilient replica-1 storage class. This helps to avoid redundant data copies and enables resilient management at the application level.

Recovering to a replacement cluster for Metro-DR

When a primary or a secondary cluster of Metro-DR fails, the cluster can be either repaired or wait for the recovery of the existing cluster, or replace the cluster entirely if the cluster is irredeemable. Fusion Data Foundation provides the ability to replace a failed primary or a secondary cluster with a new cluster and enable failover (relocate) to the new cluster.

Red Hat OpenShift virtualization workloads for Metro-DR

Metropolitan disaster recovery (Metro-DR) solution can be easily set up for OpenShift Virtualization workloads using Fusion Data Foundationn.

Support setting of RBD storage class as the default

The Ceph RADOS block device (RBD) storage class can be set as the default storage class during the deployment of Fusion Data Foundation on bare metal and IBM Power platforms. This helps to avoid manual annotation of the storage cluster when it is required to set Ceph RBD as the default storage class. In addition, it helps to avoid the confusion of selecting the correct storage class.

Performance profiles

Fusion Data Foundation provides an option to choose a resource profile based on the availability of resources during deployment. The performance profile helps to obtain enhanced performance levels. The following performance profiles can be configured both during deployment and post deployment:

- Lean - To be used 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 - To be used when recommended resources are available. This profile provides a balance between resource consumption and performance for diverse workloads.
- Performance - To be used 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.

Ability to create backing stores in Red Hat OpenShift cluster that use AWS Security Token Service

Fusion Data Foundation can be deployed on an OpenShift cluster that has the Amazon Web Services security token service (AWS STS) enabled and then backing stores of type `aws-sts-s3` can be created using the Multicloud Object Gateway command-line interface.

Runbooks for Fusion Data Foundation alerts

Fusion Data Foundation alerts include runbooks that provide guidance to fix problems on clusters that are surfaced by alerts. Alerts displayed in Fusion Data Foundation have links to the corresponding runbooks.

Allow expansion of encrypted RBD volumes

With this release, the expansion of the encrypted RADOS block device (RBD) volume feature is generally available. This feature provides resize capability for encrypted RBD persistent volume claims (PVCs).

Improved cluster availability with additional monitor daemon components

Fusion Data Foundation provides the ability to configure up to five Ceph monitor daemon components in an internal mode deployment based on the number of racks or zones when there are three, five, or more number of failure domains present in the deployment. Ceph monitor count can be increased to improve the availability of the cluster.

Alerts for monitoring system overload

Fusion Data Foundation introduces three new alerts to monitor the system that is getting overloaded. The new alerts are `OSDCPUloadHigh`, `MDSCPUUsageHigh`, and `MDSCacheUsageHigh`. These alerts improve the visibility to the current system performance and suggest tuning it when needed.

Shallow volumes support for snapshot or clone

With this release, PVC creation from snapshot functionality in Fusion Data Foundation supports shallow volumes. These shallow volumes act as a reference to the source subvolume snapshot with no actual new subvolume being created in CephFS. The supported access mode for the shallow volume is `ReadOnlyMany`. When such PVCs are mounted, it means that the respective CephFS subvolume snapshot is exposed to the workloads. These shallow volumes help to reduce the time and resources to create clones.

Note: It is not possible to take a snapshot of the ROX PVC and creating a ROX PVC clone from ROX PVC results in a pending state. This is an expected behavior.

Support for Logical Partition (LPAR) deployment

Fusion Data Foundation on IBM Z/VM supports Logical Partition (LPAR) as one of the additional deployment methods.

Enhancements

This section describes the major enhancements introduced in IBM Storage Fusion Data Foundation 4.15.

- [Deployment of one active and one standby MGR pods by OCS Operator](#)
- [Support for custom timeouts for Reclaim Space operation](#)
- [Modularized must-gather utility](#)
- [Prehook to MCG's database pod to gracefully flush caches when the pod is going down](#)
- [All controller operations to reach one controller](#)
- [Enhanced data distribution for CephFS storage class](#)
- [Ability to use bluestore-rdr as object storage device backing store](#)

Deployment of one active and one standby MGR pods by OCS Operator

The `ocs-operator` now deploys two MGR pods by default, one active and one standby. This enhancement does not impact cluster resource requirements.

Support for custom timeouts for Reclaim Space operation

Custom timeout values can be set for the reclaim space operation to avoid the failure of the operation with the error `context deadline exceeded`. The error would occur depending on the RBD volume size and its data pattern.

Modularized must-gather utility

Fusion Data Foundation `must-gather` utility can be run in a modular mode and collect only the resources that are required. This enhancement helps to avoid long duration of time taken to run `must-gather` in some environments as well as focus on the inspected components faster.

Prehook to MCG's database pod to gracefully flush caches when the pod is going down

A prehook to Multicloud Object Gateway's database pod (DB pod) is added to gracefully flush the cache when the pod is going down. This graceful shutdown reduces the risk of corruption in the journal file of the DB when the DB pod is taken down in a planned manner. However, this is not applicable for the shutdowns through OpenShift node crash or such.

All controller operations to reach one controller

When a CSI-driver provides the `CONTROLLER_SERVICE` capability, the sidecar tries to become the leader by obtaining a lease based on the name of the CSI-driver.

The Kubernetes CSI-Addons Operator tries to connect to the random CSI-Addons sidecar that is registered and try to make the RPC calls to the random sidecar. This can create a problem if the CSI-driver has implemented some internal locking mechanism or has some local cache for the lifetime of that instance.

The NetworkFence (and other CSI-Addons) operations are only sent to a CSI-Addons sidecar that has the `CONTROLLER_SERVICE` capability. There is a single leader for the CSI-Addons sidecars that support that, and the leader can be identified by the Lease object for the `csi-drivername`.

Enhanced data distribution for CephFS storage class

This feature enables the default subvolume groups of Container Storage Interface (CSI) to be **automatically** pinned to the ranks according to the default pinning configuration. This is useful when you have multiple active CephFS metadata servers (MDSs) in the cluster. This helps to better distribute the load across MDS ranks in stable and predictable ways.

Ability to use bluestore-rdr as object storage device backing store

Fusion Data Foundation provides the ability to use `bluestore-rdr` as the object storage device (OSD) backing store for the Brownfield customers. This `bluestore-rdr` has improved performance over bluestore backend store, which is important when the cluster is required to be used for Regional Disaster Recovery (RDR). Also, it is possible to migrate the OSDs to `bluestore-rdr` from the user interface.

Technology Previews

This section describes the technology preview features introduced in IBM Storage Fusion Data Foundation 4.15 under Technology Preview support limitations.

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.

- [Multicloud Object Gateway to use external PostgreSQL](#)
- [Disaster Recovery for brownfield deployments](#)
- [Enable monitoring support for ACM Subscription application type](#)
- [Support ReadWriteOncePod access mode](#)
- [Support for efficient SELinux volume relabeling](#)

Multicloud Object Gateway to use external PostgreSQL

Multicloud Object Gateway (MCG) is allowed to use an external PostgreSQL to get a high availability solution where the PostgreSQL pod is a single point of failure. This solution provides a way to use external PostgreSQL independent of Fusion Data Foundation.

Disaster Recovery for brownfield deployments

With this release, disaster recovery can be enabled for an existing cluster which was deployed using Fusion Data Foundation 4.13 or earlier or using Fusion Data Foundation 4.14 without the Disaster Recovery flag enabled.

Enable monitoring support for ACM Subscription application type

The disaster recovery dashboard on Red Hat Advanced Cluster Management (RHACM) console is extended to display monitoring data for Subscription type applications in addition to ApplicationSet type applications.

Data such as the following can be monitored:

- Volume replication delays
- Count of protected Subscription type applications with or without replication issues,
- Number of persistent volumes with replication healthy and unhealthy
- Application-wise data like the following:

- Recovery Point Objective (RPO)
- Last sync time
- Current DR activity status (Relocating, Failing over, Deployed, Relocated, Failed Over),
- Application-wise persistent volume count with replication healthy and unhealthy.

Support ReadWriteOncePod access mode

Fusion Data Foundation introduces the new **ReadWriteOncePod** (RWOP) access mode. With this RWOP access mode, it can be ensured that only one pod across the whole cluster can read that PVC or write to it. This access mode can be used either from the YAML file or the command-line interface only.

Support for efficient SELinux volume relabeling

Fusion Data Foundation has introduced an efficient SELinux relabeling for **ReadWriteOncePod** access mode. This helps to reduce the time consumption when the volume is mounted on a remote filesystem such as CephFS and when there are many files on the volume. This operation is faster than labeling all the files and folders individually.

Hub recovery support for co-situated and neutral site deployments

The Metropolitan and Regional disaster recovery solutions of Fusion Data Foundation now support neutral site deployments and hub recovery of co-situated managed clusters using Red Hat Advanced Cluster Management. For configuring hub recovery setup, a 4th cluster is required which acts as the passive hub.

The passive hub cluster can be set up in either one of the following ways:

- 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).
- 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 respective [Configuring passive hub cluster](#) chapter for Metro-DR and [Configuring passive hub cluster](#) chapter for Regional-DR.

Bug fixes

This section describes the notable bug fixes introduced in IBM Storage Fusion Data Foundation 4.15.

- [Disaster recovery](#)
- [Multicloud Object Gateway](#)
- [Ceph](#)
- [Ceph container storage interface \(CSI\)](#)
- [Fusion Data Foundation console](#)
- [Rook](#)
- [Ceph monitoring](#)
- [Must gather](#)

Disaster recovery

Fencing takes more time than expected

Previously, fencing operations took more time than expected. This was due to reconcile of Ramen hub controller a couple of times and requeue with delay as extra checks were added to ensure that the fencing operation was complete on the managed cluster.

With this fix, the hub controller is registered for the updates in fencing state. As a result, the updates of the fencing status change is received immediately and it takes less time to finish fencing operation.

[\(BZ#2249462\)](#)

Multicloud Object Gateway

Multicloud Object Gateway failing to use the new internal certificate after rotation

Previously, Multicloud Object Gateway (MCG) client was not able to connect to S3 using the new certificate unless the MCG endpoint pods were restarted. Even though the MCG endpoint pods were loading the certificate for the S3 service at the start of the pod, the changes in the certificate were not watched, which means that rotating a certificate was not affecting the endpoint till the pods were restarted.

With this fix, a watch to check for the changes in certificate of the endpoint pods are added. As a result, the pods load the new certificate without the need for a restart.

[\(BZ#2237903\)](#)

Regenerating S3 credentials for OBC in all namespaces

Previously, the Multicloud Object Gateway command for `cbc regenerate` did not have the flag `app-namespace`. This flag is available for the other object bucket claim (OBC) operations such as creation and deletion of OBC. With this fix, the `app-namespace` flag is added to the `cbc generate` command. As a result, OBC regenerates S3 credentials in all namespaces.

[\(BZ#2242414\)](#)

Signature validation failure

Previously, in Multicloud Object Gateway, there was failure to verify signatures when operations fail as AWS's C++ software development kit (SDK) does not encode the "=" sign in signature calculations when it appears as a part of the key name.

With this fix, MCG's decoding of the path in the HTTP request is fixed to successfully verify the signature.

[\(BZ#2265288\)](#)

Postgresql DB password no longer displayed in clear text in core and endpoint logs

Previously, the internal Postgresql client in noobaa-core printed a connections parameters object to the log, and this object contained the password to connect to Postgresql DB.

With this fix, the password information is omitted from the connection object that is printed to the log, and the messages to the logs contain only the nonsensitive connection details.

Ceph

Metadata server run out of memory and reports over-sized cache

Previously, metadata server (MDS) would run out of memory as the standby-replay MDS daemons would not trim their caches.

With this fix, the MDS trims its cache when in standby-replay. As a result MDS would not run out of memory.

[\(BZ#2141422\)](#)

Ceph is inaccessible after crash or shutdown tests are run

Previously, in a stretch cluster, when a monitor is revived and is in the probing stage for other monitors to receive the latest information such as `MonitorMap` or `OSDMap`, it is unable to enter `stretch_mode`. This prevents it from correctly setting the elector's `disallowed_leaders` list, which leads to the Monitors getting stuck in `election` and Ceph eventually becomes unresponsive.

With this fix, the marked-down monitors are unconditionally added to the `disallowed_leaders` list. This fixes the problem of newly revived monitors having different `disallowed_leaders` set and getting stuck in an election.

[\(BZ#2241937\)](#)

Ceph container storage interface (CSI)

Snapshot persistent volume claim in pending state

Previously, creation of readonlymany (ROX) CephFS persistent volume claim (PVC) from snapshot source failed when a pool parameter was present in the storage class due to a bug.

With this fix, the check for the pool parameter is removed as it is not required. As a result, creation of ROX CephFS PVC from a snapshot source will be successful.

[\(BZ#2248117\)](#)

Fusion Data Foundation console

Incorrect tooltip message for the raw capacity card

Previously, the tooltip for the raw capacity card in the block pool page showed an incorrect message. With this fix, the tooltip content for the raw capacity card has been changed to display an appropriate message, "Raw capacity shows the total physical capacity from all the storage pools in the StorageSystem".

[\(BZ#2237895\)](#)

System raw capacity card not showing external mode StorageSystem

Previously, the System raw capacity card did not display Ceph external StorageSystem as the Multicloud Object Gateway (MCG) standalone and Ceph external StorageSystems were filtered out from the card.

With this fix, only the StorageSystems that do not report the total capacity as per the information reported by the `odf_system_raw_capacity_total_bytes` metric is filtered out. As a result, any StorageSystem that reports the total raw capacity is displayed on the System raw capacity card and only the StorageSystems that do not report the total capacity is not displayed in the card.

[\(BZ#2257441\)](#)

Rook

Provisioning object bucket claim with the same bucket name

Previously, for the green field use case, creation of two object bucket claims (OBCs) with the same bucket name was successful from the user interface. Even though two OBCs were created, the second one pointed to invalid credentials.

With this fix, creation of the second OBC with the same bucket name is blocked and it is no longer possible to create two OBCs with the same bucket name for green field use cases.

[\(BZ#2228785\)](#)

Change of the parameter name for the Python script used in external mode deployment

Previously, while deploying Fusion Data Foundation using Ceph storage in external mode, the Python script used to extract Ceph cluster details had a parameter name, `--cluster-name`, which could be misunderstood to be the name of the Ceph cluster. However, it represented the name of the OpenShift cluster that the Ceph administrator provided.

With this fix, the `--cluster-name` flag is changed to `--k8s-cluster-name`. The legacy flag `--cluster-name` is also supported to cater to the upgraded clusters used in automation.

[\(BZ#2244609\)](#)

Incorrect pod placement configurations while detecting Multus Network Attachment Definition CIDRS

Previously, some Fusion Data Foundation clusters failed where the network "canary" pods were scheduled on nodes without Multus cluster networks, as Fusion Data Foundation did not process pod placement configurations correctly while detecting Multus Network Attachment Definition CIDRS.

With this fix, Fusion Data Foundation was fixed to process pod placement for Multus network "canary" pods. As a result, network "canary" scheduling errors are no longer experienced.

(BZ#2249678)

Deployment strategy to avoid rook-ceph-exporter pod restart

Previously, the `rook-ceph-exporter` pod restarted multiple times on a freshly installed HCI cluster that resulted in crashing of the exporter pod and the Ceph health showing the WARN status. This was because restarting the exporter using `RollingRelease` caused a race condition resulting in crash of the exporter.

With this fix, the deployment strategy is changed to `Recreate`. As a result, exporter pods no longer crash and there is no more health WARN status of Ceph.

(BZ#2250995)

rook-ceph-rgw-ocs-storagecluster-cephobjectstore-a pod stuck in CrashLoopBackOff state

Previously, the `rook-ceph-rgw-ocs-storagecluster-cephobjectstore-a` pod was stuck in `CrashLoopBackOff` state as the RADOS Gateway (RGW) multisite zonegroup was not getting created and fetched, and the error handling was reporting wrong text.

With this release, the error handling bug in multisite configuration is fixed and fetching the zonegroup is improved by fetching it for a particular `rgw-realm` that was created earlier. As a result, the multisite configuration and `rook-ceph-rgw-ocs-storagecluster-cephobjectstore-a` pod gets created successfully.

(BZ#2253185)

Ceph monitoring

TargetDown alert reported for ocs-metrics-exporter

Previously, metrics endpoint of the `ocs-metrics-exporter` used to be unresponsive as persistent volume resync by `ocs-metrics-exporter` was blocked indefinitely.

With this fix, the blocking operations from persistent volume resync in `ocs-metrics-exporter` is removed and the metrics endpoint is responsive. Also, the `TargetDown` alert for `ocs-metrics-exporter` no longer appears.

(BZ#2168042)

Label references of object bucket claim alerts

Previously, label for the object bucket claim alerts was not displayed correctly as the format for the `label-template` was wrong. Also, a blank object bucket claim name was displayed and the description text was incomplete.

With this fix, the format is corrected. As a result, the description text is correct and complete with appropriate object bucket claim name.

(BZ#2188032)

Discrepancy in storage metrics

Previously, the capacity of a pool was reported incorrectly as a wrong metrics query was used in the Raw Capacity card in the Block Pool dashboard.

With this fix, the metrics query in the user interface is updated. As a result, the metrics of the total capacity of a block pool is reported correctly.

(BZ#2252035)

Add managedBy label to rook-ceph-exporter metrics and alerts

Previously, the metrics generated by `rook-ceph-exporter` did not have the `managedBy` label. So, it was not possible for the OpenShift console user interface to identify from which StorageSystem the metrics are generated.

With this fix, the `managedBy` label, which has the name of the StorageSystem as a value, is added through the OCS operator to the storage cluster's `Monitoring` spec. This spec is read by the Rook operator and it relabels the ceph-exporter's `ServiceMonitor` endpoint labels. As a result, all the metrics generated from this exporter will have the new label `managedBy`.

(BZ#2255491)

Must gather

Must gather logs not collected after upgrade?

Previously, the `must-gather` tool failed to collect logs after the upgrade as `Collection started <time>` was seen twice.

With this fix, the `must-gather` tool is updated to run the pre-install script only once. As a result, the tool is able to collect the logs successfully after upgrade.

(BZ#2255240)

Known issues

This section describes the known issues in IBM Storage Fusion Data Foundation 4.15.

- [Disaster recovery](#)
- [Multicloud Object Gateway](#)
- [Ceph](#)
- [Fusion Data Foundation console](#)
- [OCS operator](#)

Disaster recovery

Creating an application namespace for the managed clusters

Application namespace needs to exist on RHACM managed clusters for disaster recovery (DR) related pre-deployment actions and hence is pre-created when an application is deployed at the RHACM hub cluster. However, if an application is deleted at the hub cluster and its corresponding namespace is deleted on the managed clusters, they reappear on the managed cluster.

Workaround: `openshift-dr` maintains a namespace `manifestwork` resource in the managed cluster namespace at the RHACM hub. These resources need to be deleted after the application deletion. For example, as a cluster administrator, execute the following command on the hub cluster:

```
oc delete manifestwork -n <managedCluster namespace> <drPlacementControl name>-<namespace>-ns-mw  
(BZ#2059669)
```

ceph df reports an invalid MAX AVAIL value when the cluster is in stretch mode

When a crush rule for a Red Hat Ceph Storage cluster has multiple "take" steps, the `ceph df` report shows the wrong maximum available size for the map. The issue will be fixed in an upcoming release.

[\(BZ#2100920\)](#)

Both the DRPCs protect all the 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.

[\(BZ#2128860\)](#)

MongoDB pod is in CrashLoopBackoff because of permission errors reading data in ceph rbd volume

The OpenShift projects across different managed clusters have different security context constraints (SCC), which specifically differ in the specified UID range and/or `FSGroups`. This leads to certain workload pods and containers failing to start post failover or relocate operations within these projects, due to filesystem access errors in their logs.

Workaround: Ensure workload projects are created on all managed clusters with the same project-level SCC labels, allowing them to use the same filesystem context when failed over or relocated. Pods will no longer fail post-DR actions on filesystem-related access errors.

[\(BZ#2114573\)](#)

Disaster recovery workloads remain stuck when deleted

Workaround: When deleting a workload from a cluster, the corresponding pods might not terminate with events such as `FailedKillPod`. This might cause delay or failure in garbage collecting dependent DR resources such as the `PVC`, `VolumeReplication`, and `VolumeReplicationGroup`. It would also prevent a future deployment of the same workload to the cluster as the stale resources are not yet garbage collected.

Workaround: Reboot the worker node on which the pod is currently running and stuck in a terminating state. This results in successful pod termination and subsequently related DR API resources are also garbage collected.

[\(BZ#2159791\)](#)

When DRPolicy is applied to multiple applications under same namespace, volume replication group is not created

Workaround: When a DRPlacementControl (DRPC) is created for applications that are co-located with other applications in the namespace, the DRPC has no label selector set for the applications. If any subsequent changes are made to the label selector, the validating admission webhook in the Fusion Data Foundation Hub controller rejects the changes.

Workaround: Until the admission webhook is changed to allow such changes, the DRPC `validatingwebhookconfigurations` can be patched to remove the webhook:

```
$ oc patch validatingwebhookconfigurations vdrplacementcontrol.kb.io-lq2kz --type=json --patch='[{"op": "remove", "path": "/webhooks/fusiondatafoundationhub/v1beta1/drplacementcontrol/certified/drplacementcontrol"}]'  
(BZ#2210762)
```

Application failover hangs in FailingOver state when the managed clusters are on different versions of OpenShift Container Platform and Fusion Data Foundation

Workaround: Disaster Recovery solution with Fusion Data Foundation protects and restores persistent volume claim (PVC) data in addition to the persistent volume (PV) data. If the primary cluster is on an older Fusion Data Foundation version and the target cluster is updated to 4.15 then the failover will be stuck as the S3 store will not have the PVC data.

Workaround: When upgrading the Disaster Recovery clusters, the primary cluster must be upgraded first and then the post-upgrade steps must be run.

[\(BZ#2215462\)](#)

Failover of apps from c1 to c2 cluster hang in FailingOver

Workaround: The failover action is not disabled by Ramen when data is not uploaded to the s3 store due to s3 store misconfiguration. This means the cluster data is not available on the failover cluster during the failover. Therefore, failover cannot be completed.

Workaround: Inspect the Ramen logs after initial deployment to insure there are no s3 configuration errors reported.

```
$ oc get drpc -o yaml
```

[\(BZ#2248723\)](#)

Potential risk of data loss after hub recovery

A potential data loss risk exists following hub recovery due to an eviction routine designed to clean up orphaned resources. This routine identifies and marks **AppliedManifestWorks** instances lacking corresponding **ManifestWorks** for collection. A hardcoded grace period of one hour is provided. After this period elapses, any resources associated with the **AppliedManifestWork** become subject to garbage collection.

If the hub cluster fails to regenerate corresponding **ManifestWorks** within the initial one hour window, data loss could occur. This highlights the importance of promptly addressing any issues that might prevent the recreation of **ManifestWorks** post-hub recovery to minimize the risk of data loss.

Regional DR Cephfs based application failover show warning about subscription

After the application is failed over or relocated, the hub subscriptions show up errors stating, "Some resources failed to deploy. Use View status YAML link to view the details." This is because the application persistent volume claims (PVCs) that use CephFS as the backing storage provisioner, deployed using Red Hat Advanced Cluster Management for Kubernetes (RHACM) subscriptions, and are DR protected are owned by the respective DR controllers.

Workaround: There are no workarounds to rectify the errors in the subscription status. However, the subscription resources that failed to deploy can be checked to make sure they are PVCs. This ensures that the other resources do not have problems. If the only resources in the subscription that fail to deploy are the ones that are DR protected, the error can be ignored.

[\(BZ-2264445\)](#)

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.

[\(BZ-2264765\)](#)

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 between the 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: Shut down the monitors in one of the data centers where monitors are out of quorum (you can find this by running `ceph -s` command) and reset the connection scores of the remaining monitors.

As a result, monitors can form a quorum and Ceph becomes available again and IOs resume.

[\(Partner BZ#2265992\)](#)

Cleanup and data synchronization for ApplicationSet workloads remain stuck after older primary managed cluster is recovered post the failover

ApplicationSet based workload deployments to the 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 failed over from, rejoins the new recovered standby hub.

ApplicationSets that are disaster recovery (DR) protected and with a regional DRPolicy 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.

For a workaround, see the Troubleshooting section for Regional-DR in Configuring Fusion Data Foundation Disaster Recovery for OpenShift Workloads.

[\(BZ#2268594\)](#)

Multicloud Object Gateway

Multicloud Object Gateway instance fails to finish initialization

Due to a race in timing between the pod code run and OpenShift loading the Certificate Authority (CA) bundle into the pod, the pod is unable to communicate with the cloud storage service. As a result, default backing store cannot be created.

Workaround: Restart the Multicloud Object Gateway (MCG) operator pod:

```
$ oc delete pod noobaa-operator-<ID>
```

With the workaround the backing store is reconciled and works.

[\(BZ#2269379\)](#) and [\(BZ#2268429\)](#)

Ceph

Poor performance of the stretch clusters on CephFS

Workloads with many small metadata operations might exhibit poor performance because of the arbitrary placement of metadata server (MDS) on multi-site Data Foundation clusters.

[\(BZ#1982116\)](#)

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 the 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 different ways to workaround this issue. Depending on your business needs you can choose one of the workarounds from the knowledgebase solution <https://access.redhat.com/solutions/6221251>.

[\(Jira#3327\)](#)

Ceph reports no active mgr after workload deployment

After workload deployment, Ceph manager loses connectivity to MONs or is unable to respond to its liveness probe.

This causes the Fusion Data Foundation cluster status to report that there is "no active mgr". This causes multiple operations that use the Ceph manager for request processing to fail. For example, volume provisioning, creating CephFS snapshots, and others.

To check the status of the Fusion Data Foundation cluster, use the command `oc get cephcluster -n openshift-storage`. In the status output, the `status.ceph.details.MGR_DOWN` field will have the message "no active mgr" if your cluster has this issue.

To workaround this issue, restart the Ceph manager pods using the following commands:

```
# oc scale deployment -n openshift-storage rook-ceph-mgr-a --replicas=0  
# oc scale deployment -n openshift-storage rook-ceph-mgr-a --replicas=1
```

After running these commands, the Fusion Data Foundation cluster status reports a healthy cluster, with no warnings or errors regarding `MGR_DOWN`.

[\(BZ#2244873\)](#)

CephBlockPool creation fails when custom deviceClass is used in StorageCluster

Due to a known issue, CephBlockPool creation fails when custom deviceClass is used in StorageCluster.

[\(BZ#2248487\)](#)

Fusion Data Foundation console

Missing NodeStageVolume RPC call blocks new pods from going into Running state

NodeStageVolume RPC call is not being issued blocking some pods from going into `Running` state. The new pods are stuck in `Pending` forever.

To workaround this issue, scale down all the affected pods at once or do a node reboot. After applying the workaround, all pods should go into `Running` state.

[\(BZ#2244353\)](#)

OCS operator

Incorrect unit for the `ceph_mds_mem_rss` metric in the graph

When you search for the `ceph_mds_mem_rss` metrics in the OpenShift user interface (UI), the graphs show the y-axis in Megabytes (MB), as Ceph returns `ceph_mds_mem_rss` metric in Kilobytes (KB). This can cause confusion while comparing the results for the `MDSCacheUsageHigh` alert.

Workaround: Use `ceph_mds_mem_rss * 1000` while searching this metric in the OpenShift UI to see the y-axis of the graph in GB. This makes it easier to compare the results shown in the `MDSCacheUsageHigh` alert.

[\(BZ#2261881\)](#)

Increasing MDS memory is erasing CPU values when pods are in CLBO state

When the metadata server (MDS) memory is increased while the MDS pods are in a crash loop back off (CLBO) state, CPU request or limit for the MDS pods is removed. As a result, the CPU request or the limit that is set for the MDS changes.

Workaround: Run the `oc patch` command to adjust the CPU limits.

For example:

```
$ oc patch -n openshift-storage storagecluster ocs-storagecluster \  
--type merge \  
--patch '{"spec": {"resources": {"mds": {"limits": {"cpu": "3"},  
"requests": {"cpu": "3"}}}}}'
```

[\(BZ#2265563\)](#)

Introduction to Fusion Data Foundation

IBM Storage Fusion Data Foundation is a highly integrated collection of cloud storage and data services for Red Hat OpenShift Container Platform. It is available as part of the Red Hat OpenShift Container Platform service catalog, packaged as an operator 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 PostgreSQL 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 Storage Fusion Data Foundation provides services for, and can run internally from Red Hat OpenShift Container Platform.

- [Fusion Data Foundation Operators](#)

IBM Storage 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 easily automated.

- [Fusion Data Foundation installation overview](#)

Fusion Data Foundation consists of multiple components managed by multiple operators.

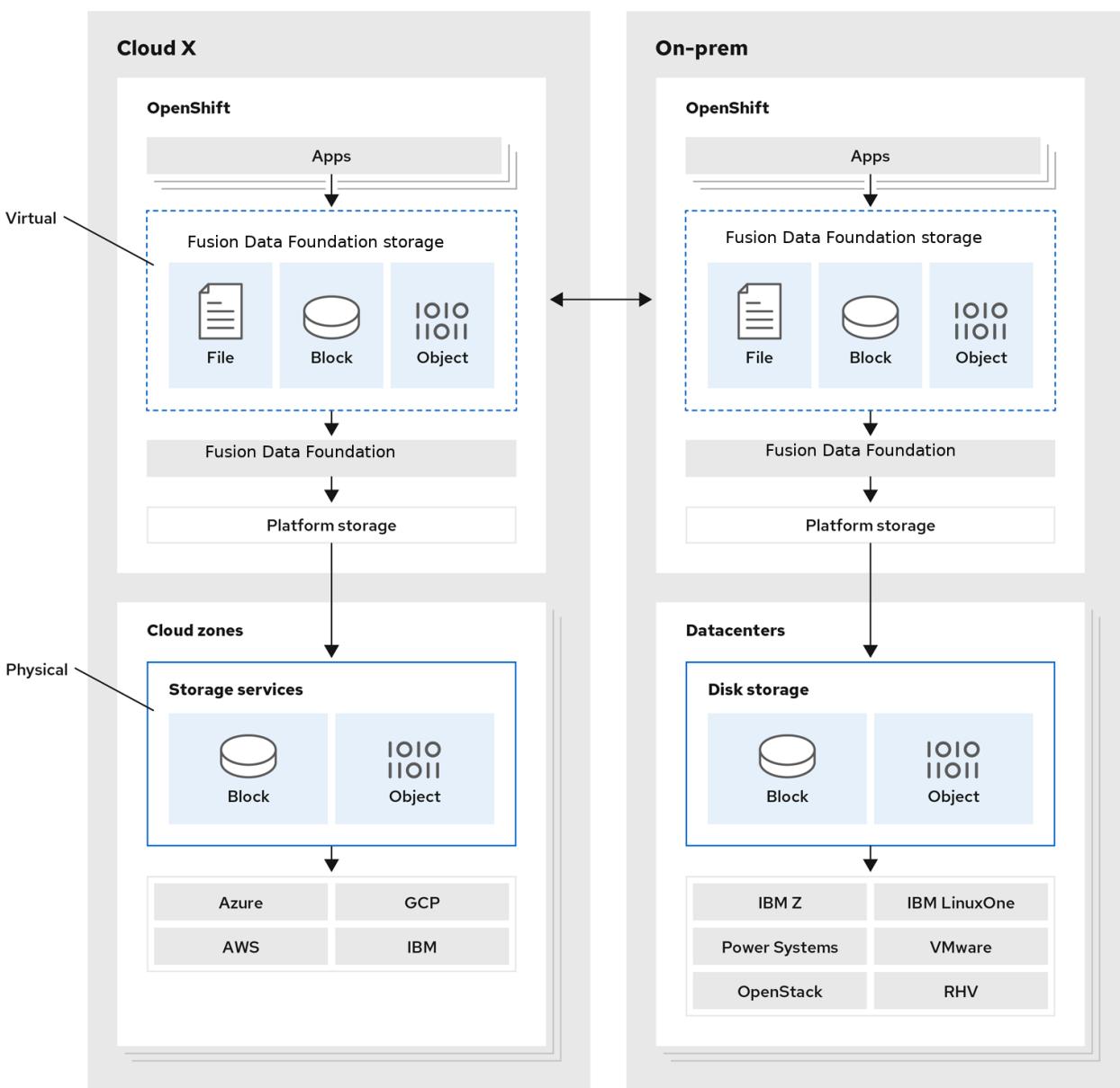
- [Fusion Data Foundation upgrade overview](#)

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).

An overview of Fusion Data Foundation architecture

IBM Storage Fusion Data Foundation provides services for, and can run internally from Red Hat OpenShift Container Platform.

Figure 1. Fusion Data Foundation architecture



171_OpenShift_1221

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 Architecture within the [Red Hat OpenShift Container Platform](#) product documentation. To know more about interoperability of components for the Fusion Data Foundation and Red Hat OpenShift Container Platform, see the [interoperability matrix](#).

For information about the architecture and lifecycle of OpenShift Container Platform, see Architecture within the [Red Hat OpenShift Container Platform](#) product documentation.

Fusion Data Foundation Operators

IBM Storage 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 easily 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 OpenShift Container Platform.

[**Responsibilities**](#)

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

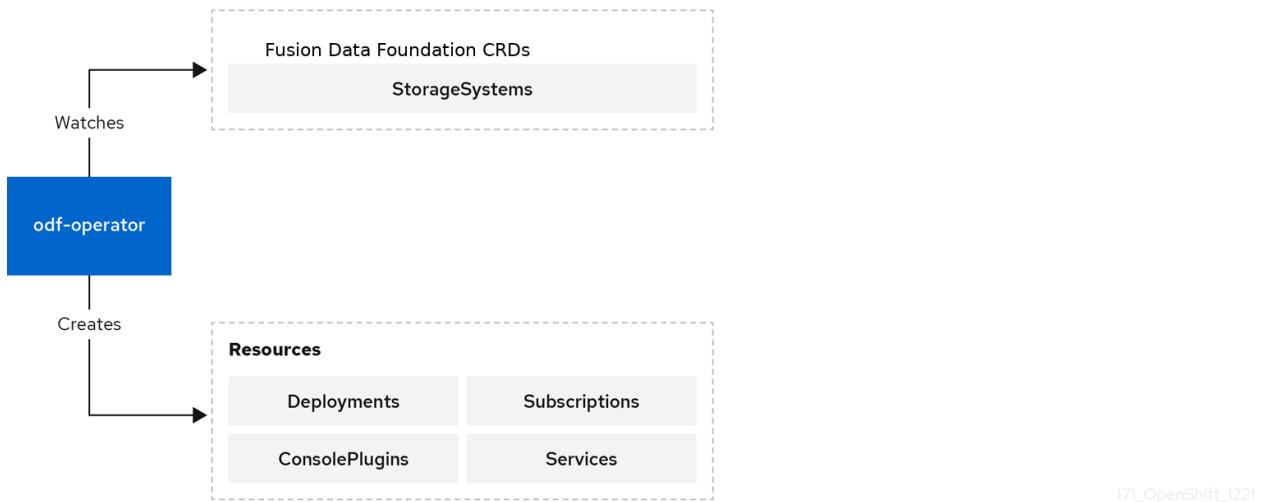
The `odf-operator` has a dependency on the `ocs-operator` package. It also manages the Subscription of the `mcg-operator`. In addition, the `odf-operator` bundle defines a second Deployment for the Fusion Data Foundation external plugin for the OpenShift Console.

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 OpenShift Container Platform.

Figure 1. Fusion Data Foundation Operator



Responsibilities

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.

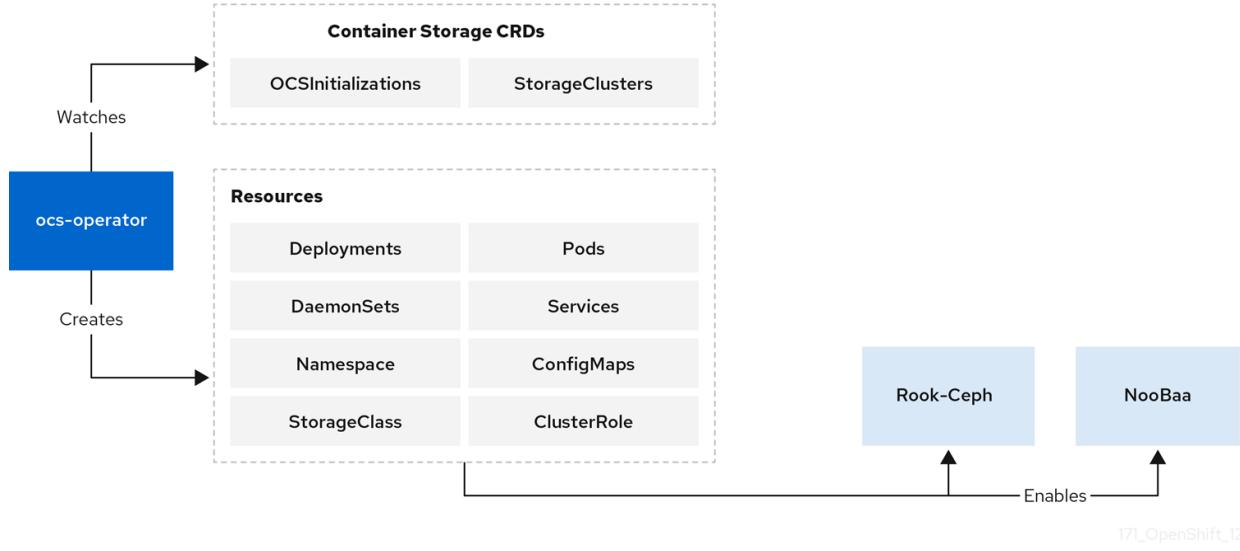
Components

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

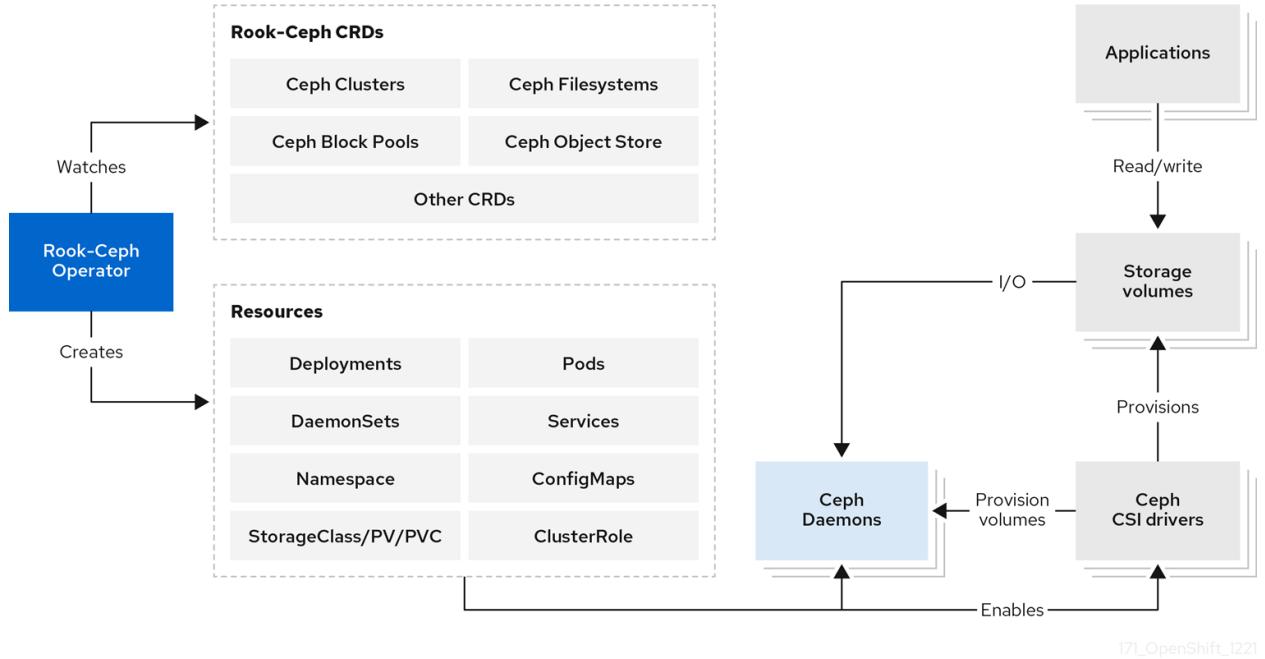
The metadata server (MDS) provides CephFS shared volumes.

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



171_OpenShift_1221

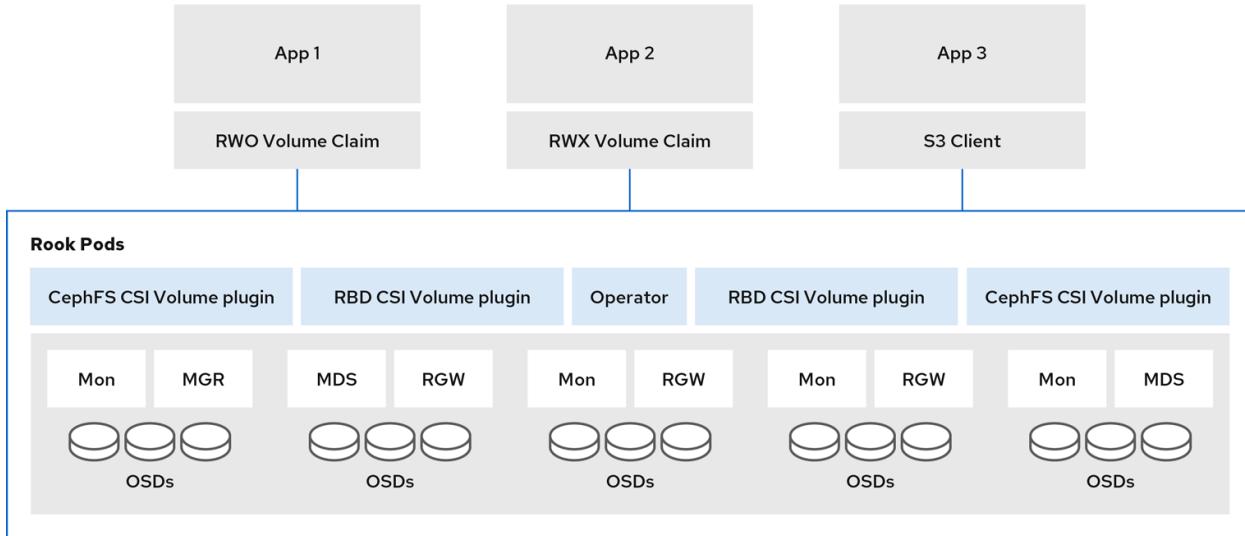
Responsibilities

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



171_OpenShift_1221

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 provisioner pods, managed by a single deployment.
- One plugin pod per node, managed by a `daemonset`.

CephFS CSI Driver

- Two provisioner pods, managed by a single deployment.
- One plugin 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](#)
- [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

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 allows you to 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](#)

- [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 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**](#)

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 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 <code>mcg-operator</code> uses the CCO to mint credentials in order to create a new GCP bucket and will create a <code>BackingStore</code> on top of that bucket.
On-prem deployment	If RGW exists, the <code>mcg-operator</code> creates a new <code>CephUser</code> and a new bucket on top of RGW and create a <code>BackingStore</code> on top of that bucket.
None of the previously mentioned deployments are applicable	The <code>mcg-operator</code> creates a <code>pv-pool</code> based on the default storage class and creates a <code>BackingStore</code> 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 <code>HorizontalPodAutoscaler</code> 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 Storage Fusion Data Foundation in external mode before Fusion deployment.

External Ceph Configuration	A <code>ConfigMap</code> that specifies the endpoints of the external <code>mons</code> .
External Ceph Credentials Secret	A <code>Secret</code> that contains the credentials to connect to the external Ceph instance.
External Ceph StorageClasses	One or more <code>StorageClasses</code> to enable the creation of volumes for RBD, CephFS, and/or RGW.
Enable CephFS CSI Driver	If a <code>CephFS StorageClass</code> is specified, configure <code>rook-ceph-operator</code> to deploy the <code>CephFS CSI Pods</code> .
Ceph RGW Configuration	If an RGW <code>StorageClass</code> 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:

<code>CephCluster</code>	Create the <code>CephCluster</code> CR to trigger Ceph reconciliation from <code>rook-ceph-operator</code> (see subsequent sections).
<code>SnapshotClasses</code>	Create the <code>SnapshotClasses</code> that applications use to create snapshots of Ceph volumes.
<code>NooBaaSystem</code>	Create the <code>NooBaa</code> CR to trigger reconciliation from <code>nocabaa-operator</code> (see subsequent sections).
<code>QuickStarts</code>	Create the <code>Quickstart</code> CRs that display the quickstart guides in the Console.

- [Cluster Creation](#)
- [NooBaa System creation](#)

Cluster Creation

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 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 <code>mcg-operator</code> uses the <code>CloudCredentialsOperator</code> (CCO) to mint credentials in order to create a new AWS:S3 bucket and creates a <code>BackingStore</code> on top of that bucket.
Microsoft Azure deployment	The <code>mcg-operator</code> uses the CCO to mint credentials in order to create a new Azure Blob and creates a <code>BackingStore</code> on top of that bucket.
Google Cloud Platform (GCP) deployment	The <code>mcg-operator</code> uses the CCO to mint credentials in order to create a new GCP bucket and will create a <code>BackingStore</code> on top of that bucket.
On-prem deployment	If RGW exists, the <code>mcg-operator</code> creates a new <code>CephUser</code> and a new bucket on top of RGW and creates a <code>BackingStore</code> on top of that bucket.
None of the previously mentioned deployments are applicable	The <code>mcg-operator</code> creates a <code>pv-pool</code> based on the default storage class and creates a <code>BackingStore</code> 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 <code>HorizontalPodAutoscaler</code> 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](#)
- [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 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 <code>mcg-operator</code> uses the <code>CloudCredentialsOperator</code> (CCO) to mint credentials in order to create a new AWS:S3 bucket and creates a <code>BackingStore</code> on top of that bucket.
Microsoft Azure deployment	The <code>mcg-operator</code> uses the CCO to mint credentials in order to create a new Azure Blob and creates a <code>BackingStore</code> on top of that bucket.
Google Cloud Platform (GCP) deployment	The <code>mcg-operator</code> uses the CCO to mint credentials in order to create a new GCP bucket and will create a <code>BackingStore</code> on top of that bucket.
On-prem deployment	If RGW exists, the <code>mcg-operator</code> creates a new <code>CephUser</code> and a new bucket on top of RGW and creates a <code>BackingStore</code> on top of that bucket.
None of the previously mentioned deployments are applicable	The <code>mcg-operator</code> creates a <code>pv-pool</code> based on the default storage class and creates a <code>BackingStore</code> 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 HorizontalPodAutoscaler 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.

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 will publish a new bundle in the IBM catalog source. The OLM will detect this and create an **InstallPlan** for the new CSV to replace the existing CSV. The Subscription approval strategy, whether Automatic or Manual, will determine whether the OLM proceeds with reconciliation or waits for administrator approval.

For Minor Version releases, Fusion Data Foundation will also publish a new bundle in the IBM catalog source. The difference is that this bundle will be part of a new channel, and channel upgrades are not automatic. The administrator must explicitly select the new release channel. Once this is done, the OLM will detect this and create an **InstallPlan** for the new CSV to replace the existing CSV. Since the channel switch is a manual operation, OLM will automatically start 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 will trigger 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 will be lost, as the new CSV will be 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](#).

Planning Fusion Data Foundation deployment

Use this information for important considerations when planning your IBM Storage 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 Storage 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.

- **[External storage services](#)**

External storage services are available for use with IBM Storage Fusion Data Foundation.

- **[Security considerations](#)**

Understand these security considerations when planning for IBM Storage Fusion Data Foundation.

- **[Subscription offerings](#)**

IBM Storage Fusion Data Foundation subscription is based on “core-pairs,” similar to Red Hat OpenShift Container Platform. The IBM Storage 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.

- **[Disconnected environment](#)**

Disconnected environment is a network restricted environment where the Operator Lifecycle Manager (OLM) cannot access the default Operator Hub and image registries, which require internet connectivity.

- **[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 Storage Fusion Data Foundation. Use this information to help you to select the most appropriate approach for your environments.

You can deploy IBM Storage 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 approach](#)**

Deployment of IBM Storage 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.

- **[External approach](#)**

IBM Storage 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 Storage 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 Storage 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 Storage 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 approach

IBM Storage 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.

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. In order 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 <code>infra</code> . For more information, see How to use dedicated worker nodes for IBM Storage Fusion Data Foundation
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. Make sure 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 Storage 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 Storage 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.

External storage services

External storage services are available for use with IBM Storage Fusion Data Foundation.

IBM Storage 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 (OBGs) 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 Storage 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 Storage 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 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 [Installing > Support for FIPS cryptography](#) within the [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 the cluster-wide proxy](#) within the [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 Storage 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 Storage 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 Storage 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.

- **[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 Storage 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.

Cluster wide encryption is supported in Fusion Data Foundation without Key Management System (KMS). Fusion Data Foundation supports with and without HashiCorp Vault KMS .

Note: Cluster wide encryption requires a validFusion 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.

Currently, HashiCorp Vault is the only supported KMS. Fusion Data Foundation supports HashiCorp Vault KV secret engine API, versions 1 and 2.

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.

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](#) > [OVN-Kubernetes network plugin](#) > [Configuring IPsec encryption within the Red Hat OpenShift Container Platform](#) product documentation.

Subscription offerings

IBM Storage Fusion Data Foundation subscription is based on “core-pairs,” similar to Red Hat OpenShift Container Platform. The IBM Storage 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 will provide 20 cores and in case of IBM Power a 2-core subscription at SMT level of 8 will provide 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 will end up consuming a full 2-core subscription once it is registered and subscribed.
- **Subscription requirements**
IBM Storage 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 will 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 table above. A large virtual machine (VM) might have 16 vCPUs, which at a SMT level 8 will require 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 will 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 will end 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 Storage 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 Storage Fusion Data Foundation should be a multiple of core-pairs.

Subscription requirements

IBM Storage 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 Storage Fusion Data Foundation](#).

Infrastructure requirements

Understand the infrastructure requirements when planning for Fusion Data Foundation.

- **[Platform requirements](#)**

IBM Storage Fusion Data Foundation 4.15 is supported only on OpenShift Container Platform version 4.15 and its next minor versions.

- **[External mode requirements](#)**

Use this information to understand external mode requirements for IBM Storage Ceph and IBM FlashSystem.

- **[Resource requirements](#)**

IBM Storage 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 Storage Fusion Data Foundation 4.15 is supported only on OpenShift Container Platform version 4.15 and its next minor versions.

Bug fixes for previous version of Fusion Data Foundation will be 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 Storage 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 Storage Fusion Data Foundation clusters only.

- **[Google Cloud](#)**

Google Cloud supports internal IBM Storage Fusion Data Foundation clusters only.

- **[Red Hat OpenStack Platform \[Technology Preview\]](#)**

Red Hat OpenStack Platform supports internal IBM Storage Fusion Data Foundation clusters and consuming external clusters.

- **[IBM Power](#)**

IBM Power supports internal IBM Storage Fusion Data Foundation clusters and consuming external clusters.

- **[IBM Z and IBM LinuxONE](#)**

IBM Z and IBM LinuxONE support internal IBM Storage Fusion Data Foundation clusters. Also, supports external mode where IBM Storage Ceph is running on x86.

Amazon EC2

Amazon EC2 supports internal IBM Storage 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 6.7, Update 2 or later
- vSphere 7.0 or later.

For more details, 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 Storage 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 Storage 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.

Google Cloud

Google Cloud supports internal IBM Storage 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 Storage 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 Storage 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 Storage 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.

External mode requirements

Use this information to understand external mode requirements for IBM Storage Ceph and IBM FlashSystem.

- **IBM Storage Ceph**

Check the supported versions and interoperability of IBM Storage Ceph with IBM Storage Fusion Data Foundation in external mode.

- **IBM FlashSystem**

To use IBM FlashSystem as a pluggable external storage on other providers, you need to first deploy it before you can deploy Fusion Data Foundation, which would use the IBM FlashSystem storage class as a backing storage.

IBM Storage Ceph

Check the supported versions and interoperability of IBM Storage Ceph with IBM Storage Fusion Data Foundation in external mode.

To check the supported versions and interoperability, open the web lab [Red Hat OpenShift Data Foundation Supportability and Interoperability Checker](#).

1. Select Service Type as ODF as Self-Managed Service.
2. Select appropriate Version from the drop down.
3. On Versions tab, click Supported RHCS versions in External Mode tab.

For instructions regarding how to install a IBM Storage Ceph cluster, see the Installing section of [IBM Storage Ceph documentation](#).

IBM FlashSystem

To use IBM FlashSystem as a pluggable external storage on other providers, you need to first deploy it before you can deploy Fusion Data Foundation, which would use the IBM FlashSystem storage class as a backing storage.

For the latest supported FlashSystem storage systems and versions, see [ODF FlashSystem driver documentation](#).

Resource requirements

IBM Storage 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

Storage Fusion Data Foundation only

Deployment Mode	Base services	Additional device Set
Internal	<ul style="list-style-type: none">• 30 CPU (logical)• 72 GiB memory• 3 storage devices	<ul style="list-style-type: none">• 6 CPU (logical)• 15 GiB memory• 3 storage devices
External	<ul style="list-style-type: none">• 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 $3 \times 10 = 30$ units of CPU are required.

For more information, see [Subscription Offerings](#) and [CPU units](#).

For additional guidance with designing your IBM Storage 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
-----------------	---------------

Deployment Mode	Base services
Internal	<ul style="list-style-type: none"> • 48 CPU (logical) • 192 GiB memory • 3 storage devices, each with additional 500GB of disk
External	<ul style="list-style-type: none"> • 24 CPU (logical) • 48 GiB memory

For example, for a three node cluster in an internal-attached devices mode deployment, a minimum of $3 \times 16 = 48$ units of CPU and $3 \times 64 = 192$ 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 will be deployed with minimum configuration when the standard deployment resource requirement is not met.

- [Compact deployment resource requirements](#)

IBM Storage 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 Storage Fusion Data Foundation only (IBM Z and IBM LinuxONE)

Deployment Mode	Base services	Additional device Set	IBM Z and LinuxONE minimum hardware requirements
Internal	<ul style="list-style-type: none"> • 30 CPU (logical) <ul style="list-style-type: none"> ◦ 3 nodes with 10 CPUs (logical) each • 72 GiB memory • 3 storage devices 	<ul style="list-style-type: none"> • 6 CPU (logical) • 15 GiB memory • 3 storage devices 	1 IFL
External	<ul style="list-style-type: none"> • 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

In order to operate a minimal cluster with 1 logical partition (LPAR), one additional IFL is required on top of the 6 IFLs. OpenShift Container Platform consumes these IFLs

Minimum deployment resource requirements

An Fusion Data Foundation cluster will be deployed with minimum configuration when the standard deployment resource requirement is not met.

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
Internal	<ul style="list-style-type: none"> • 24 CPU (logical) • 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 Storage 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	<ul style="list-style-type: none">• 24 CPU (logical)• 72 GiB memory• 3 storage devices	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 1 CPU• 4 GiB memory	<ul style="list-style-type: none">• 0.5 CPU• 4 GiB memory	<ul style="list-style-type: none">• 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.

OpenShift 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	Memory
Lean	24	72 GiB
Balanced	30	72 GiB
Performance	45	96 GiB

Important: Make sure 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 Storage 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 Storage 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 simply adding capacity or deleting or migrating content.

Dynamic storage devices

IBM Storage 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 simply adding capacity or deleting or migrating content.

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 Storage Fusion Data Foundation supports IPv6 in single stack.
 - [**Multi network plug-in \(Multus\) support**](#)
-

IPv6 support

IBM Storage 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 Storage Fusion Data Foundation also supports IPv6 auto detection and configuration. Clusters using IPv6 will automatically be configured accordingly.

OpenShift Container Platform dual stack with IBM Storage Fusion Data Foundation IPv4 is supported, however Dual stack on IBM Storage Fusion Data Foundation IPv6 is not supported.

Multi network plug-in (Multus) support

IBM Storage 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.

To use Multus, first run the Multus prerequisite validation tool. For instructions to use the tool, see [OpenShift Data Foundation - Multus.prerequisite validation tool](#). For more information about Multus networks, see [Multiple networks](#)

- [**Segregating storage traffic using Multus**](#)
 - [**When to use Multus**](#)
 - [**Multus configuration**](#)
 - [**Requirements for Multus configuration**](#)
-

Segregating storage traffic using Multus

By default, Fusion Data Foundation is configured to use the Red Hat OpenShift Software Defined Network (SDN). The default SDN carries the following types of traffic:

- Pod-to-pod traffic
- Pod-to-storage traffic, known as public network traffic when the storage is Fusion Data Foundation
- Fusion Data Foundation internal replication and rebalancing traffic, known as cluster network traffic

There are three ways to segregate Fusion Data Foundation from Red Hat OpenShift default network:

1. Reserve a network interface on the host for the public network of Fusion Data Foundation
 - Pod-to-storage and internal storage replication traffic coexist on a network that is isolated from pod-to-pod network traffic.
 - Application pods have access to the maximum public network storage bandwidth when the Fusion Data Foundation cluster is healthy.
 - When the Fusion Data Foundation cluster is recovering from failure, the application pods will have reduced bandwidth due to ongoing replication and rebalancing traffic.
2. Reserve a network interface on the host for Fusion Data Foundation's cluster network
 - Pod-to-pod and pod-to-storage traffic both continue to use OpenShift's default network.
 - Pod-to-storage bandwidth is less affected by the health of the Fusion Data Foundation cluster.
 - Pod-to-pod and pod-to-storage Fusion Data Foundation traffic might contend for network bandwidth in busy OpenShift clusters.
 - The storage internal network often has an overabundance of bandwidth that is unused, reserved for use during failures.
3. Reserve two network interfaces on the host for Fusion Data Foundation: one for the public network and one for the cluster network
 - Pod-to-pod, pod-to-storage, and storage internal traffic are all isolated, and none of the traffic types will contend for resources.
 - Service level agreements for all traffic types are more able to be ensured.
 - During healthy runtime, more network bandwidth is reserved but unused across all three networks.

Figure 1. Segregated configuration: Dual network interface

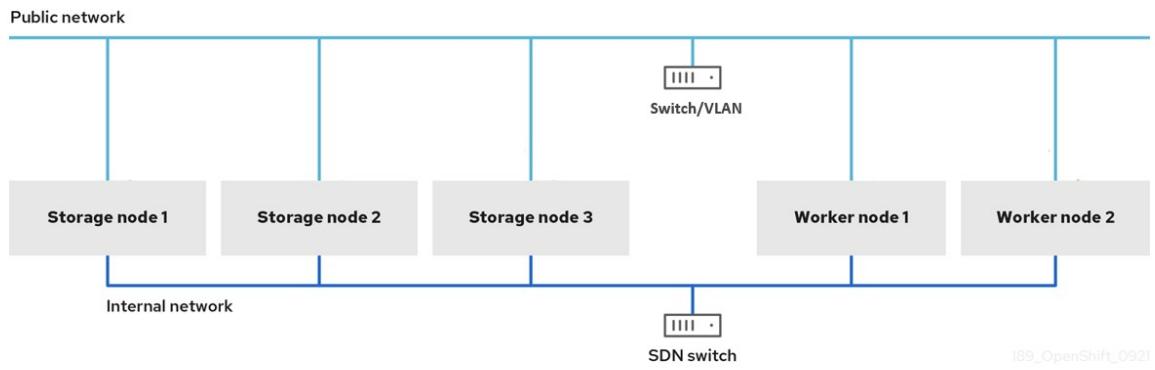
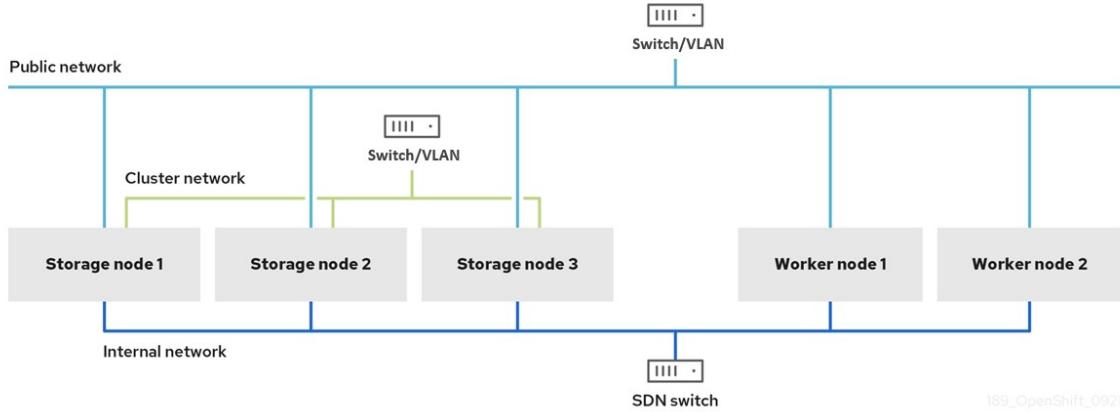


Figure 2. Full segregated configuration: Triple network interface



When to use Multus

Use Multus for Fusion Data Foundation when you need the following:

Improved latency - Multus with Fusion Data Foundation always improves latency. Use host interfaces at near-host network speeds and bypass OpenShift's software-defined Pod network. You can also perform Linux per interface level tuning for each interface.

Improved bandwidth - Dedicated interfaces for Fusion Data Foundation client data traffic and internal data traffic. These dedicated interfaces reserve full bandwidth.

Improved security - Multus isolates storage network traffic from application network traffic for added security. Bandwidth or performance might not be isolated when networks share an interface, however, you can use QoS or traffic shaping to prioritize bandwidth on shared interfaces.

Multus configuration

To use Multus, you must create network attachment definitions (NADs) before deploying the Fusion Data Foundation cluster, which is later attached to the cluster. For more information, see [Creating network attachment definitions](#).

To attach additional 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 Container Network Interface (CNI) configuration inside each of these CRs defines how that interface is created.

Fusion Data Foundation supports two types of drivers. The following tables describes the drivers and their features:

macvlan (recommended)	ipvlan
Each connection gets a sub-interface of the parent interface with its own MAC address and is isolated from the host network.	Each connection gets its own IP address and shares the same MAC address.
Uses less CPU and provides better throughput than Linux bridge or ipvlan .	L2 mode is analogous to macvlan bridge mode.
Almost always require bridge mode.	L3 mode is analogous to a router existing on the parent interface. L3 is useful for Border Gateway Protocol (BGP), otherwise use macvlan for reduced CPU and better throughput.
Near-host performance when network interface card (NIC) supports virtual ports/virtual local area networks (VLANs) in hardware.	If NIC does not support VLANs in hardware, performance might be better than macvlan .

Fusion Data Foundation supports the following two types IP address management:

whereabouts	
Uses OpenShift/Kubernetes leases to select unique IP addresses per Pod.	DHCP
Does not require a DHCP server to provide IPs for Pods.	Does not require range field. Network DHCP server can give out the same range to Multus Pods as well as any other hosts on the same network.

CAUTION:

If there is a DHCP server, ensure Multus configured IPAM does not give out the same range so that multiple MAC addresses on the network cannot have the same IP.

Requirements for Multus configuration

- The interface used for the public network must have the same interface name on each OpenShift storage and worker node, and the interfaces must all be connected to the same underlying network.
- The interface used for the cluster network must have the same interface name on each OpenShift storage node, and all the interfaces must be connected to the same underlying network. Cluster network interfaces do not have to be present on the OpenShift worker nodes.
- Each network interface used for the public or cluster network must be capable of at least 10 gigabit network speeds.
- Each network requires a separate virtual local area network (VLAN) or subnet.

For the necessary steps to configure a Multus based configuration on bare metal, see [Creating Multus networks](#).

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**
- Regional-DR**
Regional disaster recovery (Regional-DR) is composed of Red Hat Advanced Cluster Management for Kubernetes (RHACM) and IBM Storage 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 easily set up Metropolitan disaster recovery solutions for workloads based on OpenShift virtualization technology using Fusion Data Foundation. For more information, see the [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

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 [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.

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 Storage 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.

Disconnected environment

Disconnected environment is a network restricted environment where the Operator Lifecycle Manager (OLM) cannot access the default Operator Hub and image registries, which require internet connectivity.

IBM supports deployment of Fusion Data Foundation in disconnected environments where you have installed OpenShift Container Platform in restricted networks.

To install Fusion Data Foundation in a disconnected environment, see [Operators > Administrator tasks > Using Operator Lifecycle Manager on restricted networks within Red Hat OpenShift Container Platform](#) product documentation.

Note: When you install Fusion Data Foundation in a restricted network environment, apply a custom Network Time Protocol (NTP) configuration to the nodes, because by default, internet connectivity is assumed in OpenShift Container Platform and `chrony` is configured to use the `*.rhel.pool.ntp.org` servers.

For more information, see the [A newly deployed OCS 4 cluster status shows as "Degraded". Why?](#) Red Hat Knowledgebase solution on the [Red Hat Customer Portal](#) and [Installing > Installation configuration > Customizing nodes > Configuring chrony time service within the Red Hat OpenShift Container Platform](#) product documentation.

Fusion Data Foundation supports the Agent-based Installer for disconnected environment deployment. The Agent-based Installer allows you to use a mirror registry for disconnected installations. For more information, see [Installing an on-premise cluster with the Agent-based Installer](#) within the [Red Hat OpenShift Container Platform](#) product documentation.

Packages to include for Fusion Data Foundation

When you prune the `redhat-operator` index image, include the following list of packages for the Fusion Data Foundation deployment:

- `ocs-operator`
- `odf-operator`
- `mcg-operator`
- `odf-csi-addons-operator`
- `odr-cluster-operator`
- `odr-hub-operator`
- `local-storage-operator`
Only for local storage deployments.
- `odf-multicluster-orchestrator`
Only for Regional Disaster Recovery (Regional-DR) configuration.

Important: Name the CatalogSource as `redhat-operators`.

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

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 4.15 or above 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.
- The external Ceph cluster should have an existing RBD pool pre-configured for use. If it does not exist, contact your IBM Storage Ceph administrator to create one before you move ahead with Fusion Data Foundation deployment. IBM recommends to use a separate pool for each Fusion Data Foundation cluster.
- Optional: If there is a zonegroup created apart from the default zonegroup, you need to add the hostname, `rook-ceph-rgw-ocs-external-storageclustercephobjectstore.openshift-storage.svc` to the zonegroup as IBM Fusion Data Foundation sends S3 requests to the RADOS Object Gateways (RGWs) with this hostname.

Procedure

1. Install the IBM Storage Fusion Data Foundation operator.
 - a. From the Red Hat OpenShift Container Platform web management console, go to Operators > Operator Hub and search for [IBM Storage Fusion Data Foundation](#).
 - b. Click [IBM Storage 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:
 - Navigate to [Installed Operators](#) and verify that the IBM Storage Fusion Foundation Operator shows a green tick indicating successful installation.
 - Navigate 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 Storage 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 on 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, 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: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>"}}]
```

3. 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.

4. 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 Storage 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 2.8. Follow the deployment steps as documented in [Deploying IBM Storage Fusion](#).

Overview of multiple storage cluster deployments

IBM Storage 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

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 will not be 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.

Deploying Fusion Data Foundation Internal storage cluster

To deploy and verify Fusion Data Foundation storage cluster in the internal mode, refer to your respective infrastructure deployment guides.

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

- A Fusion Data Foundation cluster deployed in internal mode.
- Ensure that both the OpenShift container Platform and Fusion Data Foundation are upgraded to version 4.15.

Procedure

1. In the OpenShift web Console, navigate to Storage > Data Foundation > Storage Systems tab.
2. Click Create StorageSystem.
3. In the Backing storage page, Connect an external storage platform is selected by default.
 - a. Choose **Red Hat Ceph Storage** as the Storage platform from available options.
 - b. Click Next.
4. In the Security and Network page
 - a. Optional: To select encryption, select Enable encryption checkbox.
 - b. Click on the Download Script link to download the python script for extracting Ceph cluster details.
 - c. 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**.
 - 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 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](#).

- i. 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: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>"}}]
```

- 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) to be uploaded using the JSON file remains unchanged on the IBM Storage Ceph external cluster after the storage cluster creation.

- iv. 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
```

v. Click Browse to select and upload the JSON file.

The content of the JSON file is populated and displayed in the text box.

vi. Click Next which is enabled only after you upload the JSON file.

5. 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.

6. Click Create StorageSystem.

7. Verify the StorageSystem creation

a. Navigate to Storage > Data Foundation > Storage System tab and verify that you can view all storage clusters.

b. Verify that all components for the external Fusion Data Foundation are successfully installed. See for instructions.

What to do next

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](#)

- [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](#)

- [Verifying that storage cluster is ready](#)

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	<ul style="list-style-type: none">• <code>ocs-operator-*</code> (1 pod on any worker node)• <code>ocs-metrics-exporter-*</code> (1 pod on any worker node) Note: This pod must be present in <code>openshift-storage-extended</code> namespace as well such that there is 1 pod in each <code>openshift-storage</code> and <code>openshift-storage extended</code> namespace.• <code>odf-operator-controller-manager-*</code> (1 pod on any worker node)• <code>odf-console-*</code> (1 pod on any worker node)• <code>csi-addons-controller-manager-*</code> (1 pod on any worker node)
Rook-ceph Operator	<code>rook-ceph-operator-*</code> (1 pod on any worker node)
CSI	<ul style="list-style-type: none">• <code>cephfs</code><ul style="list-style-type: none">• <code>csi-cephfsplugin-*</code> (1 pod on each worker node)• <code>csi-cephfsplugin-provisioner-*</code> (2 pods distributed across worker nodes)• <code>rbd</code><ul style="list-style-type: none">• <code>csi-rbdplugin-*</code> (1 pod on each worker node)• <code>csi-rbdplugin-provisioner-*</code> (2 pods distributed across worker nodes)

Verifying the Fusion Data Foundation cluster 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.

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

To know more 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

To know more 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 will not be created.
- If RGW is not deployed in the external cluster, the `ocs-external-storagecluster-ceph-rgw` storage class will not be created.

For more information regarding MDS and RGW, see IBM Storage Ceph [documentation](#).

Verifying that Ceph cluster is connected

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	EXTERNAL	FSID	DATADIRHOSTPATH	MONCOUNT	AGE	PHASE	MESSAGE
ocs-external-storagecluster-cephcluster				51m	Connected		Cluster connected successfully

Verifying that storage cluster is ready

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

You can migrate application workloads from the internal mode storage classes to the external mode storage classes using [Migration Toolkit for Containers](#) using the same cluster as source and target.

Managing and allocating resources

Understand how to create, configure, and allocate storage to core services or hosted applications in IBM Storage Fusion Data Foundation.

- [**Storage classes**](#)
Use this information to understand how to create custom storage classes.
- [**Block pools**](#)
This section provides you with information on how to create, update, and delete block pools.
- [**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**](#)
- [**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 Storage 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.
- [**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**](#)
Set tolerations to bring up container storage interface (CSI) component on the nodes.
- [**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.

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 for persistent volume encryption**](#)
Persistent volume (PV) encryption guarantees 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.
- [**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.
3. Use the default Relclaim 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 will not be 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

What to do next

Once the form is completed, click Create to create the storage class.

Storage class for persistent volume encryption

Persistent volume (PV) encryption guarantees 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 . You can create an encryption enabled storage class using an external key management system (KMS) for persistent volume encryption. You need to configure access to the KMS before creating the storage class.

- [Access configuration for Key Management System \(KMS\)](#)
KMS access needs to be configured, based on your use case.
- [Creating a storage class for persistent volume encryption](#)
Create a storage class for persistent volume (PV) encryption.

Access configuration for Key Management System (KMS)

KMS access needs to be configured, based on your use case.

Configure access to KMS using one of the following ways:

- Using `vaulttokens`: allows users to authenticate using a token
- Using `vaulttenantsa` (Technology Preview): allows users to use `serviceaccounts` to authenticate with `Vault`

Important: Accessing the KMS using `vaulttenantsa` is a Technology Preview feature.

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.

- [Configuring access using vaulttokens](#)
Configure Key Management System (KMS) using `vaulttokens`, to authenticate using a token.
- [Configuring access using vaulttenantsa](#)
Configure Key Management System (KMS) using `vaulttenantsa`, allowing users to use serviceaccounts to authenticate with `Vault`.

Configuring access using `vaulttokens`

Configure Key Management System (KMS) using `vaulttokens`, to authenticate using a token.

Before you begin

- The Fusion Data Foundation cluster is in *Ready* state.
- On the external key management system (KMS):
 - Ensure that a policy with a token exists and the key value backend path in `Vault` is enabled.
 - Ensure that you are using signed certificates on your `Vault` servers.

Procedure

Create a secret in the tenant's namespace.

1. From the OpenShift Container Platform web console, go to Workloads > Secrets > Create > Key/value secret.
2. Enter Secret Name as `ceph-csi-kms-token`.
3. Enter Key as `token`.
4. Fill in the Value.
Value is the token from Vault. You can either click Browse to select and upload the file containing the token or enter the token directly in the text box.
Important: Only delete the token after all the encrypted PVCs using the `ceph-csi-kms-token` have been deleted.

5. Click Create.

Configuring access using vaulttenant

Configure Key Management System (KMS) using `vaulttenant`, allowing users to use serviceaccounts to authenticate with `Vault`.

Before you begin

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.
3. Create the following serviceaccount in the tenant namespace as shown below:

```
$ cat <<EOF | oc create -f -
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ceph-csi-vault-sa
EOF
```

About this task

Configure the Kubernetes authentication method before Fusion Data Foundation can authenticate with and start using `Vault`. These 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 Fusion Data Foundation 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

Use this sample YAML to update or create the `csi-kms-connection-detail` ConfigMap, using [Table 1](#) to configure the file.

To create a StorageClass that uses the `vaulttenant` method for PV encryption, you must either edit the existing ConfigMap or create a ConfigMap named `csi-kms-connection-details` that will hold all the information needed to establish the connection with Vault.

```
apiVersion: v1
data:
  vault-tenant-sa: |-
    {
      "encryptionKMSType": "vaulttenant",
      "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
<code>encryptionKMSType</code>	Set to <code>vaulttenant</code> to use service accounts for authentication with vault.
<code>vaultAddress</code>	The hostname or IP address of the vault server with the port number.
<code>vaultTLSServerName</code>	(Optional) The vault TLS server name.
<code>vaultAuthPath</code>	(Optional) The path where Kubernetes auth method is enabled in Vault. The default path is <code>kubernetes</code> . If the auth method is enabled in a different path other than <code>kubernetes</code> , this variable needs to be set as <code>"/v1/auth/<path>/login"</code> .
<code>vaultAuthNamespace</code>	(Optional) The Vault namespace where Kubernetes auth method is enabled.
<code>vaultNamespace</code>	(Optional) The Vault namespace where the backend path being used to store the keys exists.
<code>vaultBackendPath</code>	The backend path in Vault where the encryption keys will be stored.
<code>vaultCAFromSecret</code>	The secret in the Fusion Data Foundation cluster containing the CA certificate from Vault.
<code>vaultClientCertFromSecret</code>	The secret in the Fusion Data Foundation cluster containing the client certificate from Vault.
<code>vaultClientCertKeyFromSecret</code>	The secret in the Fusion Data Foundation cluster containing the client private key from Vault.
<code>tenantSAName</code>	(Optional) The service account name in the tenant namespace. The default value is <code>ceph-csi-vault-sa</code> . 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, you must ensure to configure access to KMS for one of the following, as detailed in [Access configuration for Key Management System \(KMS\)](#).

- Using `vaulttokens`: allows users to authenticate using a token
- Using `vaulttenant`: allows users to use `serviceaccounts` to authenticate with `Vault`

Procedure

1. Create a storage class by going to Storage-> StorageClasses and clicking Create Storage Class.

Fill in the form with the following information:

- Fill in the storage class Name and Description.
- Select the Relclaim Policy.
Use either `Delete` or `Retain`.

Note: *Delete* is the default selection.

- Select Volume binding mode.
Use either *Immediate* or *WaitForFirstConsumer*.
Note: *WaitForFirstConsumer* is the default selection.
- Select **RBD Provisioner** `openshift-storage.rbd.csi.ceph.com` which is the plugin used for provisioning the persistent volumes.
- Select Storage Pool, where the volume data is stored from the list or create a new pool.
- Select Enable encryption.
Use one of the following KMS connection details:

Select 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 menu.
- Select the Key service for the given provider from the list.

Create new KMS connection

This is applicable for `vaulttokens` only.

Select the Key Management Service Provider.

- If `Vault` is selected as the Key Management Service Provider, follow these steps:
 - 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.
 - 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.

2. Click Save.

3. Click Create.

4. 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.

From the OpenShift Web Console:

- i. Go to Storage > Storage Classes.
- ii. Click on 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": "vaultenantas",
      ...
      "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 will override 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 will override 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 on Workloads > ConfigMaps > Create ConfigMap.
3. Edit the YAML file.

The following is a sample YAML. The values to be overridden for the given tenant namespace can be specified under the `data` section as shown below:

```
---  
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.

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	kubernetes.io/aws-ebs	Delete	WaitForFirstConsumer
gp2 (default)	ebs.csi.aws.com	Delete	WaitForFirstConsumer
true			
gp2-csi			
true			

gp3-csi	104m	ebs.csi.aws.com	Delete	WaitForFirstConsumer
true	46m	openshift-storage.rbd.csi.ceph.com	Delete	WaitForFirstConsumer
ocs-storagecluster-ceph-non-resilient-rbd		openshift-storage.rbd.csi.ceph.com	Delete	Immediate
true	52m	openshift-storage.cephfs.csi.ceph.com	Delete	Immediate
ocs-storagecluster-cephfs		openshift-storage.noobaa.io/obc	Delete	Immediate
openshift-storage.noobaa.io	50m			
false				

New OSD pods are created; 3 `osd-prepare` pods and 3 additional pods.

- Verify new OSD pods are in `Running` state:

```
$ oc get pods | grep osd
```

Example output:

rook-ceph-osd-0-6dc76777bc-snbnm	2/2	Running	0	9m50s
rook-ceph-osd-1-768bdfdc4-h5n7k	2/2	Running	0	9m48s
rook-ceph-osd-2-69878645c4-bkd1q	2/2	Running	0	9m37s
rook-ceph-osd-3-64c44d7d76-zfxg9	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

Block pools

This section provides you with information on how to create, update, and delete block pools.

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.

- [**Creating a block pool**](#)

Create custom storage pools which map to storage classes.

- [**Updating an existing pool**](#)

Update existing storage pools.

- [**Deleting a pool**](#)

Use this procedure to delete a pool in Fusion Data Foundation.

Creating a block pool

Create custom storage pools which map to storage classes.

Before you begin

You must be logged into the OpenShift Container Platform web console as an administrator.

Procedure

- Click Storage > Data Foundation.
- From the Storage systems tab, select the storage system.
- Go to the BlockPools tab and then click Create Block Pool.
- Fill in the form.
 - Enter Pool name
Note: Using 2-way replication data protection policy is not supported for the default pool. However, you can use 2-way replication if you are creating an additional pool.
 - Select Data protection policy, either 2-way-Replication or 3-way-Replication.
 - 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 will not be compressed.
- Click Create.

Updating an existing pool

Update existing storage pools.

Before you begin

You must be logged into the OpenShift Container Platform web console as an administrator.

Procedure

1. Click Storage > Data Foundation.
 2. From the Storage systems tab, select the storage system.
 3. Go to the BlockPools tab and then click the Action Menu next to the pool you want to update.
 4. Click Edit Block Pool and modify the form.
Note: Using 2-way replication data protection policy is not supported for the default pool. However, you can use 2-way replication if you are creating an additional pool.
 - Change the Data protection policy, either 2-way-Replication or 3-way-Replication.
 - Change the 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 will not be compressed.
 5. Click Save.
-

Deleting a pool

Use this procedure to delete a pool in Fusion Data Foundation.

Before you begin

You must be logged into the OpenShift Container Platform web console as an administrator.

Important: A pool cannot be deleted when it is bound to a PVC. You must detach all the resources before performing this activity.

Procedure

1. Click Storage > Data Foundation.
 2. From the Storage systems tab, select the storage system.
 3. Go to the BlockPools tab and then click the Action Menu next to the pool you want to delete.
 4. Click Delete Block Pool.
 5. Confirm the removal of the pool by clicking Delete.
Note: A pool cannot be deleted when it is bound to a PVC. You must detach all the resources before performing this activity.
-

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 [Red Hat OpenShift Container Platform](#) product documentation.

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 \rightarrow 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 \rightarrow Cluster Settings \rightarrow 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 \rightarrow 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 \rightarrow 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 above, 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 \rightarrow Custom Resource Definitions.
 - b. Click the `Config` custom resource definition associated with the `image-registry.operator.openshift.io` group.
 - c. From the Instances tab, beside the cluster instance, go to Action Menu \rightarrow 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.

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-b6c8f4596-1n88h	1/1	Running	0	17d
node-ca-2nxvz	1/1	Running	0	44d
node-ca-dtwjd	1/1	Running	0	44d
node-ca-h92rzj	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 run the following command to verify if you have configured the MCG as OpenShift Image Registry backend storage successfully.

```
$ oc describe pod image-registry-b6c8f4596-1n88h
```

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:
57b943f691c878e342bac34e657b702bd6ca5488d51f839fecafa918a79a5fc6ed70184cab047601403c1f383e54d458744062dcaaa483816d82408bb56e686f
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 will 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 [Monitoring](#) > [Configuring the monitoring stack](#) > [Configuring persistent storage](#) > [Modifying retention time for Prometheus metrics data](#) within the [Red Hat OpenShift Container Platform](#) product documentation.

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 example given below 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

Project: openshift-monitoring ▾

Persistent Volume Claims

Persistent Volume Claims						
Filter by name...						
		Name	Namespace	Status	Persistent Volume	Requested
<input type="checkbox"/>	0 Pending	<input checked="" type="checkbox"/> 5 Bound	<input type="checkbox"/> 0 Lost	Select All Filters	5 Items	
	my-alertmanager-claim-alertmanager-main-0	NS	openshift-monitoring	Bound	pvc-d00428a5-0ce6-11ea-8fe8-023bdffa29edc	40Gi
	my-alertmanager-claim-alertmanager-main-1	NS	openshift-monitoring	Bound	pvc-d00be111-0ce6-11ea-8fe8-023bdffa29edc	40Gi
	my-alertmanager-claim-alertmanager-main-2	NS	openshift-monitoring	Bound	pvc-d01ac717-0ce6-11ea-8fe8-023bdffa29edc	40Gi
	my-prometheus-claim-prometheus-k8s-0	NS	openshift-monitoring	Bound	pvc-ce290f1b-0ce6-11ea-8fe8-023bdffa29edc	40Gi
	my-prometheus-claim-prometheus-k8s-1	NS	openshift-monitoring	Bound	pvc-ce361010-0ce6-11ea-8fe8-023bdffa29edc	40Gi

2. Verify that the new `alertmanager-main-*` pods appear with a state of *Running*.

- Go to Workloads > Pods.
- Click the new `alertmanager-main-*` pods to view the pod details.
- 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

Volumes					
Name	Mount Path	SubPath	Type	Permissions	Utilized By
config-volume	/etc/alertmanager/config			Read/Write	
<code>ocs-alertmanager-claim</code>	<code>/alertmanager</code>	<code>alertmanager-db</code>		Read/Write	

3. Verify that the new `prometheus-k8s-*` pods appear with a state of *Running*.

- Click the new `prometheus-k8s-*` pods to view the pod details.
- 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

Volumes					
Name	Mount Path	SubPath	Type	Permissions	Utilized By
config-out	/etc/prometheus/config_out		Container Volume	Read-only	
<code>ocs-prometheus-claim</code>	<code>/prometheus</code>	<code>prometheus-db</code>		Read/Write	

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](#).

With overprovision control, a `ClusterResourceQuota` 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 Logging within [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 will solely rely 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 Logging within [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 will be bound to a Persistent Volume Claim that requests **200GiB** of **ocs-storagecluster-ceph-rbd** storage. Each primary shard will be 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 Logging within [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 will result 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

Name	Namespace	Status	Persistent Volume	Requested
elasticsearch-elasticsearch-cdm-9rf624biv-1	openshift-logging	Bound	pvc-8993013d-1a6e-11ea-8d2f-027baef61a	200G
elasticsearch-elasticsearch-cdm-9rf624biv-2	openshift-logging	Bound	pvc-89947c90-1a6e-11ea-8d2f-027baef61a	200G
elasticsearch-elasticsearch-cdm-9rf624biv-3	openshift-logging	Bound	pvc-8995f557-1a6e-11ea-8d2f-027baef61a	200G

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-9rf624biv-3`.
 - f. Click the Persistent Volume Claim name and verify the storage class name in the PersistentVolumeClaim Overview page.

Note:

- Make sure 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 Logging within the [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

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 ReadWriteOnce (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.
Use the following command:

```
$ oc get csv $(oc get csv -n openshift-storage | grep ocs-operator | awk '{print $1}') -n openshift-storage -o jsonpath='{.metadata.annotations.external\\.features\\.ocs\\.openshift\\.io/export-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 rwo 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.xxxx", "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.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 on `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-<ast>` 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 Storage 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 Storage 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 will only schedule 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 [\[RHOC 4.x\] 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 will be used to run Fusion Data Foundation services:

```
spec:
  taints:
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      value: "true"
  metadata:
    creationTimestamp: null
  labels:
    node-role.kubernetes.io/worker: ""
    node-role.kubernetes.io/infra: ""
    cluster.ocs.openshift.io/openshift-storage: ""
```

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 ReadWriteOnce (RWO) and ReadWriteMany (RWX) access that is based on Ceph File System (CephFS) for volume mode **Filesystem**.
- PVC with ReadWriteOnce (RWO) access that is based on Ceph RADOS Block Devices (RBDs) with volume mode **Filesystem**.
- PVC with ReadWriteOnce (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.

Important: The ReadWriteOncePod (RWOP) access mode is a Technology Preview feature. 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.

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 Claims.
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

Expand Persistent Volume Claim

Increase the capacity of claim db-noobaa-db-0. This can be a time-consuming process.

Size *

50	GiB
----	-----

[Cancel](#) [Expand](#)

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. Once 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 Storage 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 Storage 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 ReadWriteOnce (RWO) and ReadWriteMany (RWX) access that is based on Ceph RBDs with volume mode **Block**
- Create a PVC with ReadWriteOnce (RWO) access that is based on Ceph RBDs with volume mode **Filesystem**
- Create a PVC with ReadWriteOnce (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.

Important: The ReadWriteOncePod (RWOP) access mode is a Technology Preview feature. 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.
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.
- [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

- 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
data-pvc	Bound	pvc-f37b8582-4b04-4676-88dd-e1b95c6abf74	1Gi	RWO	ocs-storagecluster-ceph-rbd

- 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
```

- Verify that `reclaimspacecronjob` is created in the following format:
`<pvc-name>-xxxxxx`

```
$ oc get reclaimspacecronjobs.csiaddons.openshift.io
```

For example:

NAME	SCHEDULE	SUSPEND	ACTIVE	LASTSCHEDULE	AGE
data-pvc-1642663516	@monthly				3s

- 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
```

- 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

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 1800seconds.
  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 within the 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 above `configmap` creation use the customized timeout.

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 on 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 will be 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 will be 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 on 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.
Note: For RADOS Block Device (RBD), you must select a storage class with the same pool as that of the parent PVC. Restoring the snapshot of an encrypted PVC using a storage class where encryption is not enabled and vice versa is not supported.
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.
Note: For RADOS Block Device (RBD), you must select a storage class with the same pool as that of the parent PVC. Restoring the snapshot of an encrypted PVC using a storage class where encryption is not enabled and vice versa is not supported.
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 on 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 on 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. Click Clone.
You are redirected to the new PVC details page.
6. 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

Set tolerations to bring up container storage interface (CSI) component on the nodes.

About this task

Each cluster consists of a number of dedicated nodes such as **infra** and **storage** nodes. However, an **infra** node with a custom taint will not be able to use Fusion Data Foundation Persistent Volume Claims (PVCs) on the node. So, if you want to use such nodes, you can set tolerations to bring up **csi-plugins** on the nodes. For more information, see [How to label Red Hat OpenShift Storage "nodes" as infra node?](#) on [Red Hat Customer Portal](#).

Procedure

1. Edit the configmap to add the toleration for the custom taint.
Remember to save before exiting the editor.

```
oc edit configmap rook-ceph-operator-config -n openshift-storage
```
2. Display the **configmap** to check the added toleration.

```
oc get configmap rook-ceph-operator-config -n openshift-storage -o yaml
```

Example output of the added toleration for the taint **nodetype=infra:NoSchedule**:

```
apiVersion: v1
data:
[...]
```

```

CSI_PLUGIN_TOLERATIONS: |
  - key: nodetypeoperator: Equalvalue: infraeffect: NoSchedule
    - key: node.ocp.openshift.io/storage
      operator: Equal
      value: "true"
      effect: NoSchedule
[...]
kind: ConfigMap
metadata:
[...]

```

Note: Ensure that all non-string values in the Tolerations value field has double quotation marks. For example, the values `true` which is of type boolean, and `1` which is of type int must be input as `"true"` and `"1"`.

3. Restart the `rook-ceph-operator` if the `csi-cephfsplugin-*` and `csi-rbdplugin-*` pods fail to come up on their own on the infra nodes.

```
oc delete -n openshift-storage pod <name of the rook_ceph_operator pod>
```

For example:

```
oc delete -n openshift-storage pod rook-ceph-operator-5446f9b95b-jrn2j
pod "rook-ceph-operator-5446f9b95b-jrn2j" deleted
```

What to do next

Verify that the `csi-cephfsplugin-*` and `csi-rbdplugin-*` pods are running on the `infra` nodes.

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.
- [Consuming 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}}}'
```

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 will be 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` will 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

```

- 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

```

Consuming NFS exports externally from the cluster

NFS clients outside of the cluster can mount NFS exports created by a previously-created PVC.

Procedure

- After the `nfs` flag is enabled, single-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

```

- 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

```

- 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](#).

- Get the share path from the PV.

- 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.

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.

- [**Allowing user access to the Multicloud Object Gateway Console**](#)

Allowing access to the Multicloud Object Gateway (MCG) Console to a user.

- [**Adding storage resources for hybrid or Multicloud**](#)

Understand how to add and edit storage resources for hybrid or Multicloud.

- [**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.

- [**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 easily 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](#)

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 &show
(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://a3406079515be1lea3b70683061451e-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://a340f4e1315be1lea3b70683061451e-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.

```
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 "noobaa.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-pdxdn5"
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://a3406079515be11ea3b70683061451e-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 : HKLbH1rSuVUOIsouIkSiA==

#-----#
#- S3 Addresses -#
#-----#
(1)
ExternalDNS : [https://s3-openshift-storage.apps.mycluster-cluster.qe.rh-ocs.com https://a340f4e1315be11ea3b70683061451e-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 will list 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	Partially	Returns a default value only
Bucket Notification	Not supported	
Bucket Object Versions	Supported	
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	

Allowing user access to the Multicloud Object Gateway Console

Allowing access to the Multicloud Object Gateway (MCG) Console to a user.

Before you begin

A running Fusion Data Foundation Platform.

About this task

To allow access to the Multicloud Object Gateway (MCG) Console to a user, you must be a user in the `cluster-admins` group and in the `system:cluster-admins` virtual group.

Procedure

1. To enable access to the MCG console, perform the following steps once on the cluster:
 - a. Create a `cluster-admins` group.

```
oc adm groups new cluster-admins
```

- b. Bind the group to the `cluster-admin` role.

```
oc adm policy add-cluster-role-to-group cluster-admin cluster-admins
```

2. Add or remove users from the `cluster-admins` group to control access to the MCG console.

- To add a set of users to the `cluster-admins` group, where `user-name` is the name of the user to be added:

```
oc adm groups add-users cluster-admins user-name1 user-name2 user-name3...
```

Note: If you are adding a set of users to the `cluster-admins` group, you do not need to bind the newly added users to the `cluster-admin` role to allow access to the Fusion Data Foundation dashboard.

- To remove a set of users from the `cluster-admins` group, where `user-name` is the name of the user to be removed:

```
oc adm groups remove-users cluster-admins user-name1 user-name2 user-name3...
```

What to do next

Verify the user access:

1. From the OpenShift Web Console, login as a user with access permission to Multicloud Object Gateway Console.
2. Go to Storage > Data Foundation.
3. In the **Storage Systems** tab, select the storage system and then go to Overview > Object tab.
4. Click the Multicloud Object Gateway link.
5. Click Allow selected permissions.

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 [Authentication and authorization](#), [Configuring identity providers](#), [Creating the secret within the 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:

```
$ 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:

```
$ 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
```

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/n
```

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](#)

Create an IBM COS-backed backing store using the MCG command-line interface or a YAML file to add storage resources.

- [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:

```
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

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](#)
- [Creating a new AWS STS backingstore](#)

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"
          ]
        }
      }
    }
  ]
}
```

```

1
}

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
# Make sure these values are not empty (AWS_ACCOUNT_ID, OIDC_PROVIDER)
# AWS ACCOUNT_ID is your <AWS account number>
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
# If you want to create the role before using the cluster, replace this field too.
# The OIDC provider is in the structure:
# 1) <OIDC-bucket>.s3.<aws-region>.amazonaws.com. for OIDC bucket configurations are in an S3 public bucket
# 2) `<characters>.cloudfront.net` for OIDC bucket configurations in an S3 private bucket with a public CloudFront distribution
OIDC_PROVIDER=$(oc get authentication cluster -ojson | jq -r .spec.serviceAccountIssuer | sed -e "s/^https:\/\//")
# the permission (S3 full access)
POLICYARN_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 POLICYARN; do
  echo -n "Attaching $POLICYARN ... "
  aws iam attach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "${POLICYARN}"
  echo "ok."
done <<< "$POLICYARN_STRINGS"

```

Installing OpenShift Data Foundation operator in AWS STS OpenShift cluster

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.
- Make sure that the **Update approval** field is set to **Manual**.

Creating a new AWS STS backingstore

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](#).
- Install {product-name-short} Operator. For more information, see [Installing OpenShift Data Foundation operator in AWS STS OpenShift cluster](#).

Procedure

- Install Multicloud Object Gateway (MCG).

It is installed with the default backingstore by using the short-lived credentials.

- 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:

```
$ 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 which will assume 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.
o For IBM Power, use the following command:

```
subscription-manager repos --enable=rh-odf-4-for-rhel-8-ppc64le-rpms
```

o 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:

```
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 above 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:

```
noobaa backingstore create azure-blob <backingstore_name> --account-key=<AZURE ACCOUNT KEY> --account-name=<AZURE ACCOUNT NAME> --target-blob-container <blob container name>
```

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:

```
noobaa backingstore create google-cloud-storage <backingstore_name> --private-key-json-file=<PATH TO GCP PRIVATE KEY JSON FILE>
```

```
--target-bucket <GCP bucket name>
```

backingstore_name

Name of the backingstore.

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 will be 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.

```
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. Note that 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. Note that 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.

```
noobaa backingstore create s3-compatible rgw-resource --access-key=<RGW ACCESS KEY> --secret-key=<RGW SECRET KEY> --
target-bucket=<bucket-name> --endpoint=<RGW endpoint>
```

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

- 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
```

- 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 will be 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

Name	Kind	Status	Labels	Last updated
NBC noobaa-default-bucket-class	BucketClass	Phase: Ready	app=noobaa	Jan 27, 118 pm
NBC test	BucketClass	Phase: Ready	app=noobaa	Jan 27, 119 pm

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

Name	Target Bucket	Type	Region
aws-s3-main	my-aws	AWS-S3	Eu-east-1a
bucket-main-azure	bucket-main	Azure Blob	Us-east-1b
archive-bucket	buck-1	S3 Compatibile	Us-east-1a

Name	Target Bucket	Type	Region
archive-bucket	buck-1	S3 Compatibile	Us-east-1a
data-bucket	bucket-main	Azure Blob	Us-east-1b
buck-2	buck-1	S3 Compatibile	Us-east-1a

4. Click Save.

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](#)
- [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.

Amazon S3 API endpoints for objects in namespace buckets

You can interact with objects in the namespace buckets using the Amazon Simple Storage Service (S3) API.

IBM Storage 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.

```
noobaa namespacesstore create aws-s3 <namespacesstore> --access-key <AWS ACCESS KEY> --secret-key <AWS SECRET ACCESS KEY> --target-bucket <bucket-name> -n openshift-storage
```

namespacesstore

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:

```
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:

```
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.

```
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
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.

```
noobaa namespacesstore create ibm-cos <namespacesstore> --endpoint <IBM COS ENDPOINT> --access-key <IBM ACCESS KEY> --secret-key <IBM SECRET ACCESS KEY> --target-bucket <bucket-name> -n openshift-storage
```

namespacesstore

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:

```
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:

```
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.

```
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](#)**
- **[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 namespaces.
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

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.

```
noobaa account create <noobaa-account-name> [flags]
```

For example:

```
noobaa account create testaccount --full_permission --nsfs_account_config --gid 10001 --uid 10001 --default_resource  
fs_namespacesstore
```

<code>allow_bucket_create</code>	Indicates whether the account is allowed to create new buckets. Supported values are <code>true</code> or <code>false</code> . Default value is <code>true</code> .
<code>allowed_buckets</code>	A comma-separated list of bucket names to which the user is allowed to have access and management rights.
<code>default_resource</code>	The NamespaceStore resource on which the new buckets will be created when using the S3 CreateBucket operation. The NamespaceStore must be backed by a rwx (ReadWriteMany) persistent volume claim (PVC).
<code>full_permission</code>	Indicates whether the account should be allowed full permission or not. Supported values are <code>true</code> or <code>false</code> . Default value is <code>false</code> .
<code>new_buckets_path</code>	The file system path where directories corresponding to new buckets will be created. The path is inside the file system of NamespaceStore file system PVCs where new directories are created to act as the file system mapping of newly created object bucket classes.
<code>nsfs_account_config</code>	A mandatory field that indicates if the account is used for NamespaceStore file system.
<code>nsfs_only</code>	Indicates whether the account is used only for NamespaceStore file system or not. Supported values are <code>true</code> or <code>false</code> . Default value is <code>false</code> . If it is set to 'true', it limits you from accessing other types of buckets.
<code>uid</code>	The user ID of the file system to which the MCG account will be mapped and it is used to access and manage data on the file system.
<code>gid</code>	The group ID of the file system to which the MCG account will be 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:

```
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

In order 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

- 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

```

- Navigate into the application namespace:

```
oc project <application_namespace>
```

For example:

```
oc project testnamespace
```

- 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	CAPACITY	ACCESS MODES	STORAGECLASS	AGE	STATUS	VOLUME
cephfs-write-workload-generator-no-cache-pv-claim	10Gi	RWX	ocs-storagecluster-cephfs	12s	Bound	pvc-aa58fb91-c3d2-475b-bbee-68452a613ela

```
oc get pod
```

NAME	READY	STATUS	RESTARTS	AGE
cephfs-write-workload-generator-no-cache-1-cv892	1/1	Running	0	11s

- 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 [accessing legacy application data from the openshift-storage namespace _check mountpoint](#).

```
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	STORAGECLASS	AGE	STATUS	VOLUME	CAPACITY	ACCESS MODES
cephfs-pvc-legacy		14s	Bound	cephfs-pv-legacy-openshift-storage	10Gi	RWX

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 namespacesstore:

```
noobaa namespacesstore create nsfs <nsfs_namespacesstore> --pvc-name='<cephfs_pvc_name>' --fs-backend='CEPH_FS'
nsfs_namespacesstore
A name of the NSFS namespacesstore.

cephfs_pvc_name
A name of the CephFS PVC in the openshift-storage namespace.
```

For example:

```
noobaa namespacesstore 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 namespacesstore, 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_namespacesstore>
```

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:

```
noobaa account create <user_account> --full_permission --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:

```
noobaa account create leguser --full_permission --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:

```
noobaa api bucket_api create_bucket '{
  "name": "<bucket_name>",
  "namespace": {
    "write_resource": { "resource": "<nsfs_namespacesstore>", "path": "nsfs/" },
    "read_resources": [ { "resource": "<nsfs_namespacesstore>", "path": "nsfs/" } ]
  }
}'
```

For example:

```
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_namespacesstore>
```

For example:

```
oc exec -it noobaa-endpoint-5875f467f5-546c6 -n openshift-storage -- ls -ltrZ /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 namespacesstore.

a. Delete the MCG bucket:

```
noobaa bucket delete <bucket_name>
```

For example:

```
noobaa bucket delete legacy-bucket
```

b. Delete the MCG user account.

```
noobaa account delete <user_account>
```

For example:

```
noobaa account delete leguser
```

c. Delete the NSFS namespacesstore.

```
noobaa namespacesstore delete <nsfs_namespacesstore>
```

For example:

```
noobaa namespacesstore 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 <application_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.

This allows you to 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.

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.

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.
- **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:

```
noobaa account passwd <noobaa_account_name> [options]  
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:

```
noobaa account passwd <noobaa-account-name> [flags] [options]
```

Use "noobaa options" for a list of global command-line options (applies to all commands).

Example:

```
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 : ***
```

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:

```
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:

```
noobaa account regenerate <noobaa_account_name> [options]
```

```
noobaa account regenerate
```

```
FATA[0000] ✘ Missing expected arguments: <noobaa-account-name>
```

```
Usage:
```

```
noobaa account regenerate <noobaa-account-name> [flags] [options]
```

```
Use "noobaa options" for a list of global command-line options (applies to all commands).
```

3. Once you run the **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:

```
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 **noobaa obc list** command

```
noobaa obc list
```

Example output:

NAMESPACE	NAME	BUCKET-NAME	STORAGE-CLASS	BUCKET-CLASS
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:

```
noobaa obc regenerate <bucket_claim_name> [options]
```

```
noobaa obc regenerate
```

```
FATA[0000] ✘ Missing expected arguments: <bucket-claim-name>
```

```
Usage:
```

```
noobaa obc regenerate <bucket-claim-name> [flags] [options]
```

```
Use "noobaa options" for a list of global command-line options (applies to all commands).
```

3. Once you run the `noobaa obc regenerate` command it prompts the user for confirmation. This will invalidate all connections between S3 clients and NooBaa which are connected using the current credentials.

Example:

```
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 will 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 }}'
```

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:

```
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 will be mirrored between two locations:

```
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](#)

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](#).
-

Creating a user in the Multicloud Object Gateway

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:

```
noobaa account create <noobaa-account-name> [--allow_bucket_create=true] [--allowed_buckets=[]] [--default_resource=''] [--full_permission=false]
```

`noobaa-account-name`

Specify the name of the new MCG user account.

`--allow_bucket_create`

Allows the user to create new buckets.

`--allowed_buckets`

Sets the user's allowed bucket list (use commas or multiple flags).

`--default_resource`

Sets the default resource. The new buckets are created on this default resource (including the future ones).

`--full_permission`

Allows this account to access all existing and future buckets.

Important: You need to provide permission to access at least one bucket or full permission to access all the buckets.

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 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 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.
- [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.

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:

```
noobaa abc 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:  
noobaa abc 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 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:

```
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": ""}.

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 via 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.nooCAA.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.example.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

- Use the command-line interface to generate the details of a new bucket and credentials.

```
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 will be 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: c0M0R2xVanF30DR3bHBkVW94cmY=
  AWS_SECRET_ACCESS_KEY: Wi9kcFluSWxHRzlWaFlzNk1hc0xma2JXcjM1MVhqa051S1BleXpmOQ==
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: 65117clc-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-obj
    noobaa-domain: openshift-storage.noobaa.io
  name: test2lobc
  namespace: openshift-storage
  ownerReferences:
  - apiVersion: objectbucket.io/v1alpha1
    blockOwnerDeletion: true
    controller: true
    kind: ObjectBucketClaim
    name: test2lobc
    uid: 64f04cba-f662-11e9-bc3c-0295250841af
  resourceVersion: "40752"
  selfLink: /api/v1/namespaces/openshift-storage/configmaps/test2lobc
  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.
Once 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.
Once 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:

```
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:

```
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.

```
noobaa obs 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:

```
noobaa namespacestore create ibm-cos <namespacestore> --endpoint <IBM COS ENDPOINT> --access-key <IBM ACCESS KEY> --secret-key <IBM SECRET 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:

```
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.

```
noobaa obs 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

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 easily 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.
- [**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.

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 will deploy within the cluster. If external deployment is selected, you will be provided with a YAML file to run externally.

What to do next

All nodes will be 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

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 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.

Replacing nodes

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.

- [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.

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 [/planning/platform_requirements.html](#).

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

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

- Identify the Amazon Web Service (AWS) machine instance of the node that you need to replace.
- Log in to AWS, and terminate the AWS machine instance that you identified.
- Create a new AWS machine instance with the required infrastructure.
- For more information, see [Platform requirements](#).
- Create a new OpenShift Container Platform node using the new AWS machine instance.
- Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in *Pending* state.

```
oc get csr
```

- 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>
```

- Go to Compute->Nodes and confirm that the new node is in a *Ready* state.

- 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.

- 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
```

- Workloads->Pods and confirm that at least the following pods on the new node are in a *Running* state:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

- Verify that all the other required Fusion Data Foundation pods are in *Running* state.

- 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
```

- 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:

- Create a debug pod and open a chroot environment for the one or more selected hosts:

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

- Display the list of available block devices:, using the `lsblk` command.

Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

- 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

- Log in to the OpenShift Web Console, and click Compute->Nodes.
 - Identify the faulty node that you need to replace and click on its Machine Name.
 - Go to Actions->Edit Annotations and click Add More.
 - Add `machine.openshift.io/exclude-node-draining`, and click Save.
 - 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.
- Go to Compute->Nodes and confirm that the new node is in a *Ready* state.
 - 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=""
```

- 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:

- csi-cephfsplugin-*
- csi-rbdplugin-*

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 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 [./planning/platform_requirements.html](#).

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

- Go to Action Menu > Edit Labels >.
- 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.

- 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
```

- Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

- Verify that all the other required Fusion Data Foundation pods are in *Running* state.

- 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
```

- 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:

- Create a debug pod and open a chroot environment for the one or more selected hosts:

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

- Display the list of available block devices:, using the `lsblk` command.

Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

- 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

- Log in to the OpenShift Web Console, and click Compute > Nodes.
- Identify the faulty node that you need to replace.
 Take note of its Machine Name.
- Mark the node as unschedulable, where `<node_name>` specifies the name of node that you need to replace.

```
oc adm cordon
<node_name>
```

- 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.

- Click Compute > Machines and search for the required machine.
- For the required machine, click Action menu > Delete Machine.
- Click **Delete** to confirm that the machine is deleted.
 A new machine is automatically created.
- Wait for the new machine to start and transition into *Running* state.
 Important: This activity might take at least 5 - 10 minutes or more.
- Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
- Apply the Fusion Data Foundation label to the new node using one of the following steps:

From the user interface

- Go to Action Menu > Edit Labels >.
- 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.

- 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
```
- Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
 - csi-cephfsplugin-*
 - csi-rbdplugin-*
- Verify that all the other required Fusion Data Foundation pods are in *Running* state.
- 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
```
- 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:
 - Create a debug pod and open a chroot environment for the one or more selected hosts:

```
oc debug node/<node_name>
chroot /host
```
 - Display the list of available block devices:, using the **lsblk** command.
 Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
- 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

- Identify the node and its Virtual Machine (VM) that you need replace.
- Delete the node, where **<node_name>** specifies the name of the node that needs to be replaced.

```
oc delete nodes <node_name>
```
- 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.
- Create a new VM on VMware vSphere with the required infrastructure.
 For more information, see [./planning/platform_requirements.html](#).
- Create a new OpenShift Container Platform worker node using the new VM.
- Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in *Pending* state.

```
oc get csr
```
- 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>
```
- Go to Compute > Nodes and confirm that the new node is in a *Ready* state.
- Apply the Fusion Data Foundation label to the new node using one of the following steps:

From the user interface

 - Go to Action Menu > Edit Labels >.
 - 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.

- 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
```
- Workloads > Pods and confirm that at least the following pods on the new node are in a *Running* state:
 - csi-cephfsplugin-*
 - csi-rbdplugin-*
- Verify that all the other required Fusion Data Foundation pods are in *Running* state.
- 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:

- **csi-cephfsplugin-***
- **csi-rbdplugin-***

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

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:

- `csi-cephfsplugin-*`
 - `csi-rbdplugin-*`
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.

- [**Replacing storage nodes on bare metal infrastructure**](#)
Use this information for replacing operational and failed storage nodes on a bare metal user-provisioned infrastructure.
- [**Replacing storage nodes on IBM Z or LinuxONE infrastructure**](#)
Use this information for replacing operational and failed storage nodes on an IBM Z or LinuxONE infrastructure.
- [**Replacing storage nodes on IBM Power infrastructure**](#)
Node replacement can be done proactively for an operational node, and reactively for a failed node, for the deployments related to IBM Power.
- [**Replacing storage nodes on VMware infrastructure**](#)
Use this information for replacing operational and failed storage nodes on VMware user-provisioned and installer-provisioned infrastructures.

Replacing storage nodes on bare metal infrastructure

Use this information for replacing operational and failed storage nodes on a bare metal user-provisioned infrastructure.

- To replace an operational node, see [Replacing an operational node on bare metal user-provisioned infrastructure](#).
- To replace a failed node, see [Replacing a failed node on bare metal user-provisioned infrastructure](#).
- [**Replacing an operational node on bare metal user-provisioned infrastructure**](#)
Use this procedure to replace an operational node on bare metal user-provisioned infrastructure.
- [**Replacing a failed node on bare metal user-provisioned infrastructure**](#)
Use this procedure to replace a failed node on bare metal user-provisioned infrastructure.

Replacing an operational node on bare metal user-provisioned infrastructure

Use this procedure to replace an operational node on bare metal user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node, and get the labels on the node that you need to replace, where `<node_name>` specifies the name of the node that needs to be replaced.

```
oc get nodes --show-labels | grep <node_name>
```

2. Identify the monitor pod (mon) (if any), and OSDs that are running in the node that you need to replace.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Scale down the deployments of the pods identified in the previous step.

```
oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

4. Mark the node as unschedulable

```
oc adm cordon <node_name>
```

5. Drain the node.

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

6. Delete the node.

```
oc delete node <node_name>
```

7. Get a new bare-metal machine with the required infrastructure.

For more information, see [Installing on bare metal](#).

Important: For information about how to replace a master node when you have installed Fusion Data Foundation on a three-node OpenShift compact bare-metal cluster, see the [Backup and Restore](#) guide in the OpenShift Container Platform documentation.

8. Create a new OpenShift Container Platform node using the new bare-metal machine.

9. Check for certificate signing requests (CSRs) related to Fusion Data Foundation that are in *Pending* state.

```
oc get csr
```

10. Approve all required Fusion Data Foundation CSRs for the new node, where *<certificate_name>* specifies the name of the CSR.

```
oc adm certificate approve <certificate_name>
```

11. From the OpenShift Web Console, go to Compute->Nodes and confirm that the new node is in *Ready* state.

12. Apply the Fusion Data Foundation label to the new node using one of the following steps:

From the user interface

- Go to Action Menu->Edit Labels...
- 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=""
```

13. Identify the namespace where OpenShift local storage operator is installed, and assign it to the `local_storage_project` variable.

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

14. Add a new worker node to the `localVolumeDiscovery` and `localVolumeSet`.

- Update the `localVolumeDiscovery` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]
```

Remember: Save before exiting the editor.

- Determine the `localVolumeSet` to edit.

```
oc get -n $local_storage_project localvolumeset
```

Example output:

NAME	AGE
localblock	25h

- Update the `localVolumeSet` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumeset localblock
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
```

```
- newnode.example.com  
[...]
```

Remember: Save before exiting the editor.

15. Verify that the new `localblock` Persistent Volume (PV) is available.

```
oc get pv | grep localblock | grep Available  
local-pv-551d950 512Gi RWO Delete Available  
localblock 26s
```

16. Navigate to the `openshift-storage` project.

```
oc project openshift-storage
```

17. Remove the failed OSD from the cluster.

Multiple failed OSDs can be specified, if required.

```
oc process -n openshift-storage ocs-osd-removal \  
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the `rook-ceph-osd` prefix.

You can add comma separated OSD IDs in the command to remove more than one OSD, for example, `FAILED_OSD_IDS=0,1,2`.

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.

18. 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
```

19. 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` 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
```

20. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC).

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h  
compute-1
```

If there is a PV in `Released` state, delete it:

```
oc delete pv <persistent_volume>
```

For example:

```
oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

21. Identify the `crashcollector` pod deployment.

```
oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing `crashcollector` pod deployment, delete it.

```
oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

22. 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
```

What to do next

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.

4. Ensure that the new incremental `mon` is created, and is in the *Running* state:

```
oc get pod -n openshift-storage | grep mon
```

Example output:

rook-ceph-mon-a-cd575c89b-b6k66	2/2	Running
0	38m	
rook-ceph-mon-b-6776bc469b-tzzt8	2/2	Running
0	38m	
rook-ceph-mon-d-5ff5d488b5-7v8xh	2/2	Running
0	4m8s	

OSD and monitor pod might take several minutes to get to the *Running* state.

5. 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
```

6. 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:

- Create a debug pod and open a chroot environment for the one or more selected hosts:

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

- Display the list of available block devices:, using the `lsblk` command.

Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

7. If the verification steps fail, contact [IBM Support](#).

Replacing a failed node on bare metal user-provisioned infrastructure

Use this procedure to replace a failed node on bare metal user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node, and get the labels on the node that you need to replace, where `<node_name>` specifies the name of the node that needs to be replaced.

```
oc get nodes --show-labels | grep <node_name>
```

2. Identify the monitor pod (mon) (if any), and OSDs that are running in the node that you need to replace.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Scale down the deployments of the pods identified in the previous step.

```
oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

4. Mark the node as unschedulable

```
oc adm cordon <node_name>
```

5. Remove the pods which are in *Terminating* state.

```
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 ")}'
```

6. Drain the node.

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. Delete the node.

```
oc delete node <node_name>
```

8. Get a new bare-metal machine with the required infrastructure.

For more information, see [Installing on bare metal](#).

Important: For information about how to replace a master node when you have installed Fusion Data Foundation on a three-node OpenShift compact bare-metal cluster, see the [Backup and Restore](#) guide in the OpenShift Container Platform documentation.

9. Create a new OpenShift Container Platform node using the new bare-metal machine.

10. Check for certificate signing requests (CSRs) related to Fusion Data Foundation that are in *Pending* state.

```
oc get csr
```

11. Approve all required Fusion Data Foundation CSRs for the new node, where <certificate_name> specifies the name of the CSR.

```
oc adm certificate approve <certificate_name>
```

12. From the OpenShift Web Console, go to Compute->Nodes and confirm that the new node is in *Ready* state.

13. Apply the Fusion Data Foundation label to the new node using one of the following steps:

From the user interface

- Go to Action Menu->*Edit Labels*.
- 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=""
```

14. Identify the namespace where OpenShift local storage operator is installed, and assign it to the `local_storage_project` variable.

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

15. Add a new worker node to the `localVolumeDiscovery` and `localVolumeSet`.

- Update the `localVolumeDiscovery` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]
```

Remember: Save before exiting the editor.

- Determine the `localVolumeSet` to edit.

```
oc get -n $local_storage_project localvolumeset
```

Example output:

NAME	AGE
localblock	25h

- Update the `localVolumeSet` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumeset localblock
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]
```

Remember: Save before exiting the editor.

16. Verify that the new `localblock` Persistent Volume (PV) is available.

```
oc get pv | grep localblock | grep Available
local-pv-551d950      512Gi   RWO   Delete   Available
localblock           26s
```

17. Navigate to the `openshift-storage` project.

```
oc project openshift-storage
```

18. Remove the failed OSD from the cluster.

Multiple failed OSDs can be specified, if required.

```
oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the `rook-ceph-osd` prefix.

You can add comma separated OSD IDs in the command to remove more than one OSD, for example, `FAILED_OSD_IDS=0,1,2`.

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.

19. 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
```

20. 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` 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
```

21. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC).

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h
compute-1
```

If there is a PV in `Released` state, delete it:

```
oc delete pv <persistent_volume>
```

For example:

```
oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

22. Verify that the new `localblock` Persistent Volume (PV) is available.

```
oc get pv | grep localblock | grep Available
local-pv-551d950      512Gi    RWO     Delete   Available
localblock            26s
```

23. Identify the `crashcollector` pod deployment.

```
oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing `crashcollector` pod deployment, delete it.

```
oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

24. 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
```

What to do next

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

3. Verify that all the other required Fusion Data Foundation pods are in `Running` state.

4. Ensure that the new incremental `mon` is created, and is in the `Running` state:

```
oc get pod -n openshift-storage | grep mon
```

Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66      2/2    Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8      2/2    Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh      2/2    Running
0          4m8s
```

OSD and monitor pod might take several minutes to get to the *Running* state.

- 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
```

- 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:

- Create a debug pod and open a chroot environment for the one or more selected hosts:

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

- Display the list of available block devices; using the **lsblk** command.
Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

- If the verification steps fail, contact [IBM Support](#).

Replacing storage nodes on IBM Z or LinuxONE infrastructure

Use this information for replacing operational and failed storage nodes on an IBM Z or LinuxONE infrastructure.

- To replace an operational node, see [Replacing operational nodes on IBM Z or LinuxONE infrastructure](#).
- To replace a failed node, see [Replacing failed nodes on IBM Z or LinuxONE infrastructure](#).
- Replacing operational nodes on IBM Z or LinuxONE infrastructure**
Use this procedure to replace an operational node on IBM zSystems or LinuxONE infrastructure.
- Replacing a failed node on an IBM Z infrastructure user provisioned infrastructure**
Use this procedure if you need to replace and configure failed nodes.

Replacing operational nodes on IBM Z or LinuxONE infrastructure

Use this procedure to replace an operational node on IBM zSystems or LinuxONE infrastructure.

Procedure

- Identify the node, and get the labels on the node that you need to replace, where <node_name> specifies the name of the node that needs to be replaced.

```
oc get nodes --show-labels | grep <node_name>
```

- Identify the monitor pod (mon) (if any), and OSDs that are running in the node that you need to replace.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

- Scale down the deployments of the pods identified in the previous step.

```
oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

- Mark the node as unschedulable

```
oc adm cordon <node_name>
```

- Remove the pods which are in *Terminating* state.

```
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 ")}'
```

- Drain the node.

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

- Delete the node.

```
oc delete node <node_name>
```

- Get a new IBM zSystem storage node as a replacement.

- Check for certificate signing requests (CSRs) related to Fusion Data Foundation that are in *Pending* state.

```
oc get csr
```

10. Approve all required Fusion Data Foundation CSRs for the new node, where <certificate_name> specifies the name of the CSR.

```
oc adm certificate approve <certificate_name>
```

11. From the OpenShift Web Console, go to Compute > Nodes and confirm that the new node is in *Ready* state.

12. Apply the Fusion Data Foundation label to the new node using one of the following steps:

From the user interface

- Go to Action Menu > Edit Labels >.
- 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=""
```

13. Add a new worker node to the `localVolumeDiscovery` and `localVolumeSet`.

- Update the `localVolumeDiscovery` definition to include the new node, and remove the failed node.

```
oc edit -n local-storage-project localvolumediscovery auto-discover-devices
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]
```

Remember: Save before exiting the editor.

- Determine the `localVolumeSet` to edit.

Replace `local-storage-project` with the name of your local storage project. The default project name is `openshift-local-storage` in Fusion Data Foundation.

```
oc get -n local-storage-project localvolumeset
```

Example output:

NAME	AGE
localblock	25h

- Update the `localVolumeSet` definition.

```
oc edit -n $local_storage_project localvolumeset localblock
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]
```

Remember: Save before exiting the editor.

14. Verify that the new `localblock` PV is available.

```
oc get pv | grep localblock
```

NAME	CAPA-	ACCESS	RECLAIM	STATUS	STORAGE		
					CLAIM	CLASS	AGE
local-pv-3e8964d3	931Gi	RWO	Delete	Bound	openshift-storage/ocs-deviceset-2-0	localblock	25h
					-79j94		
local-pv-414755e0	931Gi	RWO	Delete	Bound	openshift-storage/ocs-deviceset-1-0	localblock	25h
					-959rp		
local-pv-b481410	931Gi	RWO	Delete	Available		localblock	3m24s
d9c5cbd6					openshift-storage/ocs-deviceset-0-0	localblock	25h
					-nvs68		

15. Change to the `openshift-storage` project.

```
oc project openshift-storage
```

16. Remove the failed OSD from the cluster.

Multiple failed OSDs can be specified, if required.

- Identify the PVC.

The associated PVs later get deleted.

```
osd_id_to_remove=1
oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} | grep ceph.rook.io/pvc
```

In this example, `osd_id_to_remove` is the integer in the pod name immediately after the `rook-ceph-osd` prefix.

The deployment name here is `rook-ceph-osd-1`.

Example output:

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
```

In this example, the PVC name is `ocs-deviceset-localblock-0-data-0-g2mmc`.

b. Remove the failed OSD from the cluster.

```
oc process -n openshift-storage ocs-osd-removal -p FAILED OSD_IDS=${osd_id_to_remove} | oc create -f -
```

You can remove more than one OSD by adding comma separated OSD IDs in the command. (For example: FAILED OSD_IDS=0,1,2)

Warning: This step results in OSD being completely removed from the cluster. Ensure that the correct value of `osd_id_to_remove` is provided.

Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

17. Verify that the OSD was removed successfully by checking the status of the `ocs-osd-removal` pod.

A status of *Completed* confirms that the OSD removal job succeeded.

```
oc get pod -l job-name=ocs-osd-removal-osd_id_to_remove -n openshift-storage
```

Note: If `ocs-osd-removal` 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-osd_id_to_remove -n openshift-storage
--tail=-1
```

It may be necessary to manually cleanup the removed OSD as follows:

```
ceph osd crush remove osd.osd_id_to_remove
ceph osd rm osd.osd_id_to_remove
ceph auth del osd.osd_id_to_remove
ceph osd crush rm osd.osd_id_to_remove
```

18. Delete the PV associated with the failed node.

a. Identify the PV associated with the PVC.

The PVC name must be identical to the name that is obtained while removing the failed OSD from the cluster.

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

```
local-pv-5c9b8982 500Gi RWO Delete Released openshift-storage/ocs-deviceset-localblock-0-data-0-g2mmc localblock 24h work
```

b. If there is a PV in *Released* state, delete it.

```
oc delete pv <persistent-volume>
```

For example:

```
oc delete pv local-pv-5c9b8982
persistentvolume "local-pv-5c9b8982" deleted
```

19. Identify the `crashcollector` pod deployment.

```
oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing `crashcollector` pod deployment, delete it.

```
oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

20. Delete the `ocs-osd-removal` job.

```
oc delete job ocs-osd-removal-${osd_id_to_remove}
```

Example output:

```
job.batch "ocs-osd-removal-0" deleted
```

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

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 node on an IBM Z infrastructure user provisioned infrastructure

Use this procedure if you need to replace and configure failed nodes.

Before you begin

- IBM recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOC) cluster.

Procedure

- Identify the node and get labels on the node to be replaced. Make a note of the rack label.

```
oc get nodes --show-labels | grep <node_name>
```

- Identify the mon (if any) and object storage device (OSD) pods that are running in the node to be replaced.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

- Scale down the deployments of the pods identified in the previous step.

For example:

```
oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

- Mark the node as unschedulable.

```
oc adm cordon <node_name>
```

- Remove the pods which are in Terminating state.

```
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 \"")}'
```

- Drain the node.

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

- Delete the node.

```
oc delete node <node_name>
```

- Get a new IBM Z infrastructure machine with required infrastructure.

- Create a new OpenShift Container Platform node using the new IBM Z infrastructure machine.

- Check for certificate signing requests (CSRs) related to Fusion Data Foundation that are in *Pending* state:

```
oc get csr
```

- Approve all required Fusion Data Foundation CSRs for the new node:

```
oc adm certificate approve <Certificate_Name>
```

- Click Compute > Nodes in OpenShift Web Console, confirm if the new node is in Ready state.

- Apply the Fusion Data Foundation label to the new node using any one of the following:

From User interface

- For the new node, click Action Menu > Edit Labels

- Add `cluster.ocs.openshift.io/openshift-storage` and click Save.

From Command line interface

- Execute the following command to apply the Fusion Data Foundation label to the new node:

```
oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

- Add a new worker node to `localVolumeDiscovery` and `localVolumeSet`.

- Update the `localVolumeDiscovery` definition to include the new node and remove the failed node.

```

oc edit -n local-storage-project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
  nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:

```

```

    - server1.example.com
    - server2.example.com
    #- server3.example.com
    - newnode.example.com
[...]

```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

b. Determine which `localVolumeSet` to edit.

Replace `local-storage-project` in the following commands with the name of your local storage project. The default project name is `openshift-local-storage` in Fusion Data Foundation 4.6 and later. Previous versions use `local-storage` by default.

```

oc get -n local-storage-project localvolumeset
NAME      AGE
localblock 25h

```

c. Update the `localVolumeSet` definition to include the new node and remove the failed node.

```

oc edit -n local-storage-project localvolumeset localblock
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]

```

Remember to save before exiting the editor.

In the above example, `server3.example.com` was removed and `newnode.example.com` is the new node.

15. Verify that the new `localblock` PV is available.

```

oc get pv | grep localblock
          CAPA- ACCESS RECLAIM                      STORAGE
          NAME   CITY MODES POLICY STATUS   CLAIM           CLASS   AGE
local-pv- 931Gi RWO Delete Bound   openshift-storage/
3e8964d3          ocs-deviceset-2-0             localblock 25h
                           -79j94
local-pv- 931Gi RWO Delete Bound   openshift-storage/
414755e0          ocs-deviceset-1-0             localblock 25h
                           -959rp
local-pv- 931Gi RWO Delete Available   localblock 3m24s
b481410
local-pv- 931Gi RWO Delete Bound   openshift-storage/
d9c5cbd6          ocs-deviceset-0-0             localblock 25h
                           -nvs68

```

16. Change to the `openshift-storage` project.

```
oc project openshift-storage
```

17. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

```

oc process -n openshift-storage ocs-osd-removal \
-p FAILED OSD IDS=failed-osd-id1,failed-osd-id2 | oc create -f -

```

18. Verify that the OSD was removed successfully by checking the status of the `ocs-osd-removal` 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
```

19. 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 `ocs-osd-removal` 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
```

20. Delete the PV associated with the failed node.

a. Identify the PV associated with the PVC.

```

oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h
compute-1

```

b. Delete the PV.

```
oc delete pv <persistent-volume>
```

For example:

```
oc delete pv local-pv-d6bf175b
persistentvolume "local-pv-d9c5cbd6" deleted
```

21. Delete the `crashcollector` pod deployment.

```
oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

22. Delete the `ocs-osd-removal` job.

```
oc delete job ocs-osd-removal-${osd_id_to_remove}
```

Example output:

```
job.batch "ocs-osd-removal-0" deleted
```

What to do next

1. Execute the following command and 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. Click Workloads > Pods, confirm that at least the following pods on the new node are in **Running** state:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

3. Verify that all other required Fusion Data Foundation pods are in **Running** state.

Make sure that the new incremental `mon` is created and is in the **Running** state.

```
oc get pod -n openshift-storage | grep mon
```

Example output:

<code>rook-ceph-mon-c-64556f7659-c2ngc</code>	1/1	Running	0	6h14m
<code>rook-ceph-mon-d-7c8b74dc4d-tt6hd</code>	1/1	Running	0	4h24m
<code>rook-ceph-mon-e-57fb8c657-wg5f2</code>	1/1	Running	0	162m

OSD and Mon might take several minutes to get to the **Running** state.

4. Verify that new OSD pods are running on the replacement node.

```
oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. Optional: 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 previous step, do the following:

- Create a debug pod and open a chroot environment for the selected host(s).

```
oc debug node/<node name>
$ chroot /host
```

- Run “`lsblk`” and check for the “crypt” keyword beside the `ocs-deviceset` name(s)

```
$ lsblk
```

6. If verification steps fail, [contact Red Hat Support](#).

Replacing storage nodes on IBM Power infrastructure

Node replacement can be done proactively for an operational node, and reactively for a failed node, for the deployments related to IBM Power.

- [Replacing an operational or failed storage node on IBM Power](#)

Use this procedure to replace an operational or failed node on IBM Power.

Before you begin

- Ensure that the replacement nodes are configured with the similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

- Identify the node, and get the labels on the node that you need to replace, where `<node_name>` specifies the name of the node that needs to be replaced.

```
oc get nodes --show-labels | grep <node_name>
```

2. Identify the monitor pod (mon) (if any), and OSDs that are running in the node that you need to replace.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Scale down the deployments of the pods identified in the previous step.

```
oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

4. Mark the node as unschedulable

```
oc adm cordon <node_name>
```

5. Remove the pods which are in *Terminating* state.

```
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 ")}'
```

6. Drain the node.

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. Delete the node.

```
oc delete node <node_name>
```

8. Get a new IBM Power machine with the required infrastructure.
For more information, see [Installing a cluster on IBM Power](#).

9. Create a new OpenShift Container Platform node using the new IBM Power machine.

10. Check for certificate signing requests (CSRs) related to Fusion Data Foundation that are in *Pending* state.

```
oc get csr
```

11. Approve all required Fusion Data Foundation CSRs for the new node, where <certificate_name> specifies the name of the CSR.

```
oc adm certificate approve <certificate_name>
```

12. From the OpenShift Web Console, go to Compute > Nodes and confirm that the new node is in *Ready* state.

13. Apply the Fusion Data Foundation label to the new node using one of the following steps:

From the user interface

- Go to Action Menu > Edit Labels >.
- 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=""
```

14. Identify the namespace where OpenShift local storage operator is installed, and assign it to the `local_storage_project` variable.

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

15. Add a newly added worker node to the `localVolume`.

- Determine the `localVolume` you need to edit.

```
oc get -n $local_storage_project localvolume
```

Example output:

NAME	AGE
localblock	25h

- Update the `localVolume` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolume localblock
```

Example output:, where `worker-0` is removed and `worker-3` is the new node.

```
[...]
  nodeSelector:
  nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
            # worker-0
            - worker-1
            - worker-2
```

```
- worker-3  
[...]
```

Remember: Save before exiting the editor.

16. Verify that the new `localblock` Persistent Volume (PV) is available.

```
oc get pv | grep localblock
```

Example output:

NAME	CAPACITY	ACCESSMODES	RECLAIMPOLICY	STATUS	CLAIM	STORAGECLASS	AGE
local-pv-3e8964d3	500Gi	RWO	Delete	Bound	ocs-deviceset-localblock-2-data-0-mdbg9	localblock	25h
local-pv-414755e0	500Gi	RWO	Delete	Bound	ocs-deviceset-localblock-1-data-0-4cs1f	localblock	25h
local-pv-b481410	500Gi	RWO	Delete	Available		localblock	3m24s
local-pv-5c9b8982	500Gi	RWO	Delete	Bound	ocs-deviceset-localblock-0-data-0-g2mmc	localblock	25h

17. Navigate to the `openshift-storage` project.

```
oc project openshift-storage
```

18. Remove the failed OSD from the cluster.

Multiple failed OSDs can be specified, if required.

- a. Identify the Persistent Volume Claim (PVC).

```
osd_id_to_remove=  
oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} | grep ceph.rook.io/pvc  
<osd_id_to_remove>  
The integer in the pod name immediately after the rook-ceph-osd prefix.
```

In this example, the deployment name is `rook-ceph-osd-1`.

Example output:

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc  
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
```

- b. Remove the failed OSD from the cluster.

Multiple failed OSDs can be specified, if required.

```
oc process -n openshift-storage ocs-osd-removal \  
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -  
  
<failed_osd_id>  
Is the integer in the pod name immediately after the rook-ceph-osd prefix.
```

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.

Important: This step results in the OSD being completely removed from the cluster. Ensure that the correct value of `osd_id_to_remove` is provided.

19. 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
```

20. 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` 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
```

21. Delete the PV associated with the failed node.

- a. Identify the PV associated with the PVC.

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-5c9b8982 500Gi RWO Delete Released openshift-storage/ocs-deviceset-localblock-0-data-0-g2mmc  
localblock 24h worker-0
```

The PVC name must be identical to the name that is obtained while removing the failed OSD from the cluster.

- b. If there is a PV in `Released` state, delete it.

```
oc delete pv <persistent_volume>
```

For example:

```
oc delete pv local-pv-5c9b8982
```

Example output:

```
persistentvolume "local-pv-5c9b8982" deleted
```

22. Identify the `crashcollector` pod deployment.

```
oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing `crashcollector` pod deployment, delete it.

```
oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

23. 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
```

What to do next

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.

4. Ensure that the new incremental `mon` is created and is in the *Running* state.

```
oc get pod -n openshift-storage | grep mon
```

Example output:

rook-ceph-mon-b-74f6dc9d6-411zq	1/1	Running	0	6h14m
rook-ceph-mon-c-74948755c-h7wtx	1/1	Running	0	4h24m
rook-ceph-mon-d-598f69869b-4bv49	1/1	Running	0	162m

The OSD and monitor pod might take several minutes to get to the *Running* state.

5. 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
```

6. 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.

7. If the verification steps fail, contact [IBM Support](#).

Replacing storage nodes on VMware infrastructure

Use this information for replacing operational and failed storage nodes on VMware user-provisioned and installer-provisioned infrastructures.

- To replace an operational node, see:
 - [Replacing an operational node on VMware user-provisioned infrastructure](#).
 - [Replacing an operational node on VMware installer-provisioned infrastructure](#).

- To replace a failed node, see:
 - [Replacing a failed node on VMware user-provisioned infrastructure](#).
 - [Replacing a failed node on VMware installer-provisioned infrastructure](#).

- [**Replacing an operational node on VMware user-provisioned infrastructure**](#)

Use this procedure to replace an operational node on VMware user-provisioned infrastructure.

- [**Replacing an operational node on VMware installer-provisioned infrastructure**](#)

Use this procedure to replace an operational node on VMware installer-provisioned infrastructure.

- [**Replacing a failed node on VMware user-provisioned infrastructure**](#)

Use this procedure to replace a failed node on VMware user-provisioned infrastructure.

- [**Replacing a failed node on VMware installer-provisioned infrastructure**](#)

Use this procedure to replace a failed node on VMware installer-provisioned infrastructure.

Replacing an operational node on VMware user-provisioned infrastructure

Use this procedure to replace an operational node on VMware user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node, and get the labels on the node that you need to replace, where <node_name> specifies the name of the node that needs to be replaced.

```
oc get nodes --show-labels | grep <node_name>
```

2. Identify the monitor pod (mon) (if any), and OSDs that are running in the node that you need to replace.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Scale down the deployments of the pods identified in the previous step.

```
oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage  
oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage  
oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

4. Mark the node as unschedulable

```
oc adm cordon <node_name>
```

5. Drain the node.

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

6. Delete the node.

```
oc delete node <node_name>
```

7. Log in to VMware vSphere and terminate the Virtual Machine (VM) that you have identified.

8. Create a new VM on VMware vSphere with the required infrastructure.

For more information see, [Infrastructure requirements](#).

9. Create a new OpenShift Container Platform worker node using the new VM.

10. Check for certificate signing requests (CSRs) related to Fusion Data Foundation that are in *Pending* state.

```
oc get csr
```

11. Approve all required Fusion Data Foundation CSRs for the new node, where <certificate_name> specifies the name of the CSR.

```
oc adm certificate approve <certificate_name>
```

12. From the OpenShift Web Console, go to Compute->Nodes and confirm that the new node is in *Ready* state.

13. 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=""
```

14. Identify the namespace where OpenShift local storage operator is installed, and assign it to the `local_storage_project` variable.

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)  
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

15. Add a new worker node to the `localVolumeDiscovery` and `localVolumeSet`.

- a. Update the `localVolumeDiscovery` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]  
  nodeSelector:  
    nodeSelectorTerms:  
      - matchExpressions:
```

```

    - key: kubernetes.io/hostname
      operator: In
      values:
      - server1.example.com
      - server2.example.com
      #- server3.example.com
      - newnode.example.com
[...]

```

Remember: Save before exiting the editor.
 b. Determine the `localVolumeSet` to edit.

```
oc get -n $local_storage_project localvolumeset
```

Example output:

NAME	AGE
localblock	25h

c. Update the `localVolumeSet` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumeset localblock
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```

[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
  - key: kubernetes.io/hostname
    operator: In
    values:
    - server1.example.com
    - server2.example.com
    #- server3.example.com
    - newnode.example.com
[...]

```

Remember: Save before exiting the editor.

16. Verify that the new `localblock` Persistent Volume (PV) is available.

```
oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

17. Navigate to the `openshift-storage` project.

```
oc project openshift-storage
```

18. Remove the failed OSD from the cluster.

Multiple failed OSDs can be specified, if required.

```
oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

`<failed_osd_id>`
 Is the integer in the pod name immediately after the `rook-ceph-osd` prefix.

You can add comma separated OSD IDs in the command to remove more than one OSD, for example, `FAILED_OSD_IDS=0,1,2`.

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.

19. 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
```

20. 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` 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
```

21. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC).

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h
compute-1
```

If there is a PV in *Released* state, delete it:

```
oc delete pv <persistent_volume>
```

For example:

```
oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

22. Identify the `crashcollector` pod deployment.

```
oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing `crashcollector` pod deployment, delete it.

```
oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

23. 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
```

What to do next

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.

4. Ensure that the new incremental `mon` is created, and is in the *Running* state:

```
oc get pod -n openshift-storage | grep mon
```

Example output:

rook-ceph-mon-a-cd575c89b-b6k66	2/2	Running
0	38m	
rook-ceph-mon-b-6776bc469b-tzzt8	2/2	Running
0	38m	
rook-ceph-mon-d-5ff5d488b5-7v8xh	2/2	Running
0	4m8s	

OSD and monitor pod might take several minutes to get to the *Running* state.

5. 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
```

6. 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:

- Create a debug pod and open a chroot environment for the one or more selected hosts:

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

- Display the list of available block devices:, using the `lsblk` command.
Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

7. If the verification steps fail, contact [IBM Support](#).

Replacing an operational node on VMware installer-provisioned infrastructure

Use this procedure to replace an operational node on VMware installer-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

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. Identify the node, and get the labels on the node that you need to replace, where <node_name> specifies the name of the node that needs to be replaced.

```
oc get nodes --show-labels | grep <node_name>
```

4. Identify the monitor pod (mon) (if any), and OSDs that are running in the node that you need to replace.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

5. Scale down the deployments of the pods identified in the previous step.

```
oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

6. Mark the node as unschedulable

```
oc adm cordon <node_name>
```

7. Drain the node.

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

8. Click Compute > Machines and search for the required machine.

9. For the required machine, click Action menu > Delete Machine.

10. Click **Delete** to confirm that the machine is deleted.

A new machine is automatically created.

11. Wait for the new machine to start and transition into *Running* state.

Important: This activity might take at least 5 - 10 minutes or more.

12. Go to Compute > Nodes and confirm that the new node is in a *Ready* state.

13. Physically add a new device to the node.

14. Apply the Fusion Data Foundation label to the new node using one of the following steps:

From the user interface

- Go to Action Menu > Edit Labels >.
- 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=""
```

15. Identify the namespace where OpenShift local storage operator is installed, and assign it to the `local_storage_project` variable.

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

16. Add a new worker node to the `localVolumeDiscovery` and `localVolumeSet`.

a. Update the `localVolumeDiscovery` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]
```

Remember: Save before exiting the editor.

b. Determine the `localVolumeSet` to edit.

```
oc get -n $local_storage_project localvolumeset
```

Example output:

NAME	AGE
localblock	25h

c. Update the `localVolumeSet` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumeset localblock
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]
```

Remember: Save before exiting the editor.

17. Verify that the new **localblock** Persistent Volume (PV) is available.

```
oc get pv | grep localblock | grep Available
local-pv-551d950  512Gi   RWO   Delete   Available
localblock        26s
```

18. Navigate to the **openshift-storage** project.

```
oc project openshift-storage
```

19. Remove the failed OSD from the cluster.

Multiple failed OSDs can be specified, if required.

```
oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>
Is the integer in the pod name immediately after the **rook-ceph-osd** prefix.

You can add comma separated OSD IDs in the command to remove more than one OSD, for example, **FAILED_OSD_IDS=0,1,2**.

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.

20. 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
```

21. 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** 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
```

22. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC).

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b  1490Gi   RWO   Delete   Released   openshift-storage/ocs-deviceset-0-data-0-6c5pw   localblock   2d22h
compute-1
```

If there is a PV in *Released* state, delete it:

```
oc delete pv <persistent_volume>
```

For example:

```
oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

23. Identify the **crashcollector** pod deployment.

```
oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing **crashcollector** pod deployment, delete it.

```
oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

24. 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
```

What to do next

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:

- csi-cephfsplugin-*
- csi-rbdplugin-*

3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.

4. Ensure that the new incremental mon is created, and is in the *Running* state:

```
oc get pod -n openshift-storage | grep mon
```

Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66      2/2     Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8      2/2     Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh      2/2     Running
0          4m8s
```

OSD and monitor pod might take several minutes to get to the *Running* state.

5. 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
```

6. 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.

7. If the verification steps fail, contact [IBM Support](#).

Replacing a failed node on VMware user-provisioned infrastructure

Use this procedure to replace a failed node on VMware user-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node, and get the labels on the node that you need to replace, where <node_name> specifies the name of the node that needs to be replaced.

```
oc get nodes --show-labels | grep <node_name>
```

2. Identify the monitor pod (mon) (if any), and OSDs that are running in the node that you need to replace.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Identify the monitor pod (mon) (if any), and OSDs that are running in the node that you need to replace.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

4. Mark the node as unschedulable

```
oc adm cordon <node_name>
```

5. Remove the pods which are in *Terminating* state.

```
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 ")}'
```

6. Drain the node.

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. Delete the node.

```
oc delete node <node_name>
```

8. Log in to VMware vSphere and terminate the Virtual Machine (VM) that you have identified.
9. Create a new VM on VMware vSphere with the required infrastructure.
For more information see, [Infrastructure requirements](#).
10. Create a new OpenShift Container Platform worker node using the new VM.
11. Check for certificate signing requests (CSRs) related to Fusion Data Foundation that are in *Pending* state.

```
oc get csr
```

12. Approve all required Fusion Data Foundation CSRs for the new node, where <certificate_name> specifies the name of the CSR.

```
oc adm certificate approve <certificate_name>
```

13. From the OpenShift Web Console, go to Compute > Nodes and confirm that the new node is in *Ready* state.
14. 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=""
```

15. Identify the namespace where OpenShift local storage operator is installed, and assign it to the `local_storage_project` variable.

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

16. Add a new worker node to the `localVolumeDiscovery` and `localVolumeSet`.

- a. Update the `localVolumeDiscovery` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]
```

Remember: Save before exiting the editor.

- b. Determine the `localVolumeSet` to edit.

```
oc get -n $local_storage_project localvolumeset
```

Example output:

NAME	AGE
localblock	25h

- c. Update the `localVolumeSet` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumeset localblock
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - server1.example.com
              - server2.example.com
              #- server3.example.com
              - newnode.example.com
[...]
```

Remember: Save before exiting the editor.

17. Verify that the new `localblock` Persistent Volume (PV) is available.

```
oc get pv | grep localblock | grep Available
```

local-pv-551d950	512Gi	RWO	Delete	Available
localblock	26s			

18. Navigate to the `openshift-storage` project.

```
oc project openshift-storage
```

19. Remove the failed OSD from the cluster.

Multiple failed OSDs can be specified, if required.

```
oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the `rook-ceph-osd` prefix.

You can add comma separated OSD IDs in the command to remove more than one OSD, for example, `FAILED_OSD_IDS=0,1,2`.

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.

20. 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
```

21. 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` 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
```

22. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC).

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h
compute-1
```

If there is a PV in `Released` state, delete it:

```
oc delete pv <persistent_volume>
```

For example:

```
oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

23. Identify the `crashcollector` pod deployment.

```
oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing `crashcollector` pod deployment, delete it.

```
oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

24. 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
```

What to do next

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

3. Verify that all the other required Fusion Data Foundation pods are in `Running` state.

4. Ensure that the new incremental `mon` is created, and is in the `Running` state:

```
oc get pod -n openshift-storage | grep mon
```

Example output:

```

rook-ceph-mon-a-cd575c89b-b6k66      2/2    Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8      2/2    Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh      2/2    Running
0          4m8s

```

OSD and monitor pod might take several minutes to get to the *Running* state.

- 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> | grep osd
```

- 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:

- Create a debug pod and open a chroot environment for the one or more selected hosts:

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

- Display the list of available block devices; using the **lsblk** command.

Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

- If the verification steps fail, contact [IBM Support](#).

Replacing a failed node on VMware installer-provisioned infrastructure

Use this procedure to replace a failed node VMware installer-provisioned infrastructure.

Before you begin

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

- Log in to the OpenShift Web Console, and click Compute->Nodes.

- Identify the faulty node that you need to replace.

Take note of its Machine Name.

- Identify the node, and get the labels on the node that you need to replace, where **<node_name>** specifies the name of the node that needs to be replaced.

```
oc get nodes --show-labels | grep <node_name>
```

- Identify the monitor pod (mon) (if any), and OSDs that are running in the node that you need to replace.

```
oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

- Scale down the deployments of the pods identified in the previous step.

```
oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

- Mark the node as unschedulable

```
oc adm cordon <node_name>
```

- Remove the pods which are in *Terminating* state.

```
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 \"")}'
```

- Drain the node.

```
oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

- Click Compute->Machines and search for the required machine.

10. For the required machine, click Action menu->Delete Machine.

11. Click **Delete** to confirm that the machine is deleted.

A new machine is automatically created.

12. Wait for the new machine to start and transition into *Running* state.

Important: This activity might take at least 5 - 10 minutes or more.

13. Go to Compute->Nodes and confirm that the new node is in a *Ready* state.

14. Physically add a new device to the node.

15. Apply the Fusion Data Foundation label to the new node using one of the following steps:

From the user interface

- Go to Action Menu->Edit Labels->.
- 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=""
```

16. Identify the namespace where OpenShift local storage operator is installed, and assign it to the `local_storage_project` variable.

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)  
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

17. Add a new worker node to the `localVolumeDiscovery` and `localVolumeSet`.

- a. Update the `localVolumeDiscovery` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]  
  nodeSelector:  
    nodeSelectorTerms:  
      - matchExpressions:  
          - key: kubernetes.io/hostname  
            operator: In  
            values:  
              - server1.example.com  
              - server2.example.com  
              #- server3.example.com  
              - newnode.example.com  
[...]
```

Remember: Save before exiting the editor.

- b. Determine the `localVolumeSet` to edit.

```
oc get -n $local_storage_project localvolumeset
```

Example output:

NAME	AGE
localblock	25h

- c. Update the `localVolumeSet` definition to include the new node, and remove the failed node.

```
oc edit -n $local_storage_project localvolumeset localblock
```

Example output, where `server3.example.com` is removed, and `newnode.example.com` is the new node:

```
[...]  
  nodeSelector:  
    nodeSelectorTerms:  
      - matchExpressions:  
          - key: kubernetes.io/hostname  
            operator: In  
            values:  
              - server1.example.com  
              - server2.example.com  
              #- server3.example.com  
              - newnode.example.com  
[...]
```

Remember: Save before exiting the editor.

18. Verify that the new `localblock` Persistent Volume (PV) is available.

```
oc get pv | grep localblock | grep Available  
local-pv-551d950  512Gi   RWO   Delete   Available  
localblock        26s
```

19. Navigate to the `openshift-storage` project.

```
oc project openshift-storage
```

20. Remove the failed OSD from the cluster.

Multiple failed OSDs can be specified, if required.

```
oc process -n openshift-storage ocs-osd-removal \  
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the `rook-ceph-osd` prefix.

You can add comma separated OSD IDs in the command to remove more than one OSD, for example, `FAILED_OSD_IDS=0,1,2`.

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.

21. 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
```

22. 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` 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
```

23. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC).

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h  
compute-1
```

If there is a PV in *Released* state, delete it:

```
oc delete pv <persistent_volume>
```

For example:

```
oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

24. Identify the `crashcollector` pod deployment.

```
oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing `crashcollector` pod deployment, delete it.

```
oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

25. 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
```

What to do next

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

3. Verify that all the other required Fusion Data Foundation pods are in *Running* state.

4. Ensure that the new incremental `mon` is created, and is in the *Running* state:

```
oc get pod -n openshift-storage | grep mon
```

Example output:

rook-ceph-mon-a-cd575c89b-b6k66	2/2	Running
0	38m	
rook-ceph-mon-b-6776bc469b-tzzt8	2/2	Running
0	38m	
rook-ceph-mon-d-5fff5d488b5-7v8xh	2/2	Running
0	4m8s	

OSD and monitor pod might take several minutes to get to the *Running* state.

5. 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
```

6. 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:

- Create a debug pod and open a chroot environment for the one or more selected hosts:

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

- Display the list of available block devices:, using the `lsblk` command.

Check for the `crypt` keyword beside the one or more `ocs-deviceset` names.

7. If the verification steps fail, contact [IBM Support](#).

Replacing devices

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.

- **[Dynamically provisioned Fusion Data Foundation deployed using local storage devices](#)**

When replacing a device in a dynamically created storage cluster on an Azure installer-provisioned infrastructure, you must replace the storage device. This can be done on either an operational or a failed local storage device.

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 [./planning/platform_requirements.html](#).

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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

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:

- csi-cephfsplugin-*
- csi-rbdplugin-*

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:

- csi-cephfsplugin-*
- csi-rbdplugin-*

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> | grep 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

- Go to Action Menu->Edit Labels->.
- 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:

- `csi-cephfsplugin-*`
- `csi-rbdplugin-*`

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> | grep 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:

<code>rook-ceph-osd-0-6d77d6c7c6-m8xj6</code>	0/1	<code>CrashLoopBackOff</code>	0	24h	10.129.0.16	<code>compute-2</code>	<code><none></code>
<code>rook-ceph-osd-1-85d99fb95f-2svc7</code>	1/1	<code>Running</code>	0	24h	10.128.2.24	<code>compute-0</code>	<code><none></code>
<code>rook-ceph-osd-2-6c66cdb977-jp542</code>	1/1	<code>Running</code>	0	24h	10.130.0.18	<code>compute-1</code>	<code><none></code>
<code><none></code>							

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 will be `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 contain data.  
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. Navigate 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 | grep -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 above command gets stuck due to insufficient privileges, run the following commands:

1. Press `CTRL+Z` to exit the above 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
    rook-ceph-osd-1-85d99fb95f-2svc7
    rook-ceph-osd-2-6c66cdb977-jp542
```

	1/1	Running	0	4m47s
	1/1	Running	0	1d20h
	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
5m	Bound	pvc-7c9bcf7-de68-40e1-95f9-0b0d7c0ae2fc	512Gi	RWO	thin	
1d20h	Bound	pvc-9e7e00cb-6b33-402e-9dc5-b8df4fd9010f	512Gi	RWO	thin	
1d20h	Bound	pvc-38cdfceea7e-42a5-a6e1-aaa6d4924291	512Gi	RWO	thin	

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:

- csi-cephfsplugin-*
- csi-rbdplugin-*

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:

- csi-cephfsplugin-*
- csi-rbdplugin-*

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](#)

Dynamically provisioned Fusion Data Foundation deployed using local storage devices

When replacing a device in a dynamically created storage cluster on an Azure installer-provisioned infrastructure, you must replace the storage device. This can be done on either an operational or a failed local storage device.

- [Replacing operational or failed storage devices on clusters backed by local storage devices](#)
Use this information to replace operational or failed storage devices on clusters backed by local storage devices.
- [Replacing operational or failed storage devices on IBM Power](#)
Use this information to replace operational or failed storage devices on IBM Power infrastructure.
- [Replacing operational or failed storage devices on IBM Z or LinuxONE infrastructure](#)
Use this information to replace operational or failed storage devices on IBM Z or LinuxONE infrastructure.

Replacing operational or failed storage devices on clusters backed by local storage devices

Use this information to replace operational or failed storage devices on clusters backed by local storage devices.

Before you begin

- It is recommended that the replacement devices are configured with similar infrastructure and resources to the device being replaced.
- 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

You can replace an object storage device (OSD) in Fusion Data Foundation deployed using local storage devices on the following infrastructures:

- Bare metal
- VMware
- Red Hat Virtualization

Note: One or more underlying storage devices may need to be replaced.

Procedure

1. Remove the underlying storage device from relevant worker node.
2. Verify that relevant OSD Pod has moved to `CrashLoopBackOff` state.
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:

<code>rook-ceph-osd-0-6d77d6c7c6-m8xj6</code>	0/1	<code>CrashLoopBackOff</code>	0	24h	10.129.0.16	<code>compute-2</code>	<code><none></code>
<code><none></code>							
<code>rook-ceph-osd-1-85d99fb95f-2svc7</code>	1/1	<code>Running</code>	0	24h	10.128.2.24	<code>compute-0</code>	<code><none></code>
<code><none></code>							
<code>rook-ceph-osd-2-6c66cdb977-jp542</code>	1/1	<code>Running</code>	0	24h	10.130.0.18	<code>compute-1</code>	<code><none></code>
<code><none></code>							

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.

3. 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
```

- 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 `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 contain
pod "rook-ceph-osd-0-6d77d6c7c6-m8xj6" force deleted
```

- Remove the old OSD from the cluster so that you can add a new OSD.

- Delete any old `ocs-osd-removal` jobs.

```
oc delete -n openshift-storage job ocs-osd-removal-job
job.batch "ocs-osd-removal-job" deleted
```

- Navigate to the `openshift-storage` project.

```
oc project openshift-storage
```

- 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.

- 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
```

- 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
```

- 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.

- 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"
```

- For each of the previously identified nodes, do the following:

- Create a `debug` pod and `chroot` to the host on the storage node, where `<node name>` is the name of the node.

```
oc debug node/<node name>
```

```
$ chroot /host
```

- Find a relevant device name based on the PVC names identified in the previous step, where `<pvc name>` is the name of the PVC..

```
dmsetup ls| grep <pvc name>
```

Example output:

```
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt (253:0)
```

- Remove the mapped device. Where `<ocs-deviceset-name>` is the name of the relevant device based on the PVC names identified in the previous step.

```
$ cryptsetup luksClose --debug --verbose <ocs-deviceset-name>
```

Important: If the above command gets stuck due to insufficient privileges, run the following commands:

- Press `CTRL+Z` to exit the above command.

- 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
```

9. Find the persistent volume (PV) that need to be deleted.

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock
```

10. Delete the PV.

```
oc delete pv <pv_name>
```

11. Physically add a new device to the node.

12. Track the provisioning of PVs for the devices that match the `deviceInclusionSpec`. It can take a few minutes to provision the PVs.

```
oc -n openshift-local-storage describe localvolumeset localblock
```

Example output:

```
[...]
Status:
  Conditions:
    Last Transition Time: 2020-11-17T05:03:32Z
    Message: DiskMaker: Available, LocalProvisioner: Available
    Status: True
    Type: DaemonSetsAvailable
    Last Transition Time: 2020-11-17T05:03:34Z
    Message: Operator reconciled successfully.
    Status: True
    Type: Available
  Observed Generation: 1
  Total Provisioned Device Count: 4
Events:
  Type Reason Age From Message
  ---- ---- -- -- -----
  Normal Discovered 2m30s (x4 localvolumeset- node.example.com -
           NewDevice over 2m30s) symlink-controller found possible
                   matching disk,
                   waiting 1m to claim

  Normal FoundMatch 89s (x4 localvolumeset- node.example.com -
           ingDisk over 89s) symlink-controller symlinkning matching
                           disk
```

Once the PV is provisioned, a new OSD pod is automatically created for the PV.

13. Delete the `ocs-osd-removal` job(s).

```
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

Important: If the new OSD does not show as `Running` after a few minutes, restart the `rook-ceph-operator` pod to force a reconciliation.

```
oc delete pod -n openshift-storage -l app=rook-ceph-operator
```

Example output:

```
pod "rook-ceph-operator-6f74fb5bff-2d982" deleted
```

2. Verify that a new PVC is created.

```
oc get -n openshift-storage pvc | grep localblock
```

Example output:

ocs-deviceset-0-0-c2mqb	Bound	local-pv-b481410	1490Gi	RWO	localblock	5m
ocs-deviceset-1-0-959rp	Bound	local-pv-414755e0	1490Gi	RWO	localblock	1d20h
ocs-deviceset-2-0-79j94	Bound	local-pv-3e8964d3	1490Gi	RWO	localblock	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.

```
oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-pod-name>
```

where <OSD-pod-name> is the name of the OSD pod.

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).

```
oc debug node/<node name>
```

where <node name> is the name of the node.

```
$ 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.

Note: A full data recovery may take longer depending on the volume of data being recovered.

Replacing operational or failed storage devices on IBM Power

Use this information to replace operational or failed storage devices on IBM Power infrastructure.

Before you begin

- It is recommended that replacement devices are configured with similar infrastructure and resources to the device being replaced.
- 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

You can replace an object storage device (OSD) in Fusion Data Foundation deployed using local storage devices on IBM Power.

Note: One or more underlying storage devices may need to be replaced.

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-0-86bf8cdc8-4nb5t 0/1 crashLoopBackOff 0 24h 10.129.2.26 worker-0 <none> <none>
rook-ceph-osd-1-7c99657cfb-jdzvz 1/1 Running 0 24h 10.128.2.46 worker-1 <none> <none>
rook-ceph-osd-2-5f9f6dfb5b-2mnw9 1/1 Running 0 24h 10.131.0.33 worker-2 <none> <none>
```

In this example, `rook-ceph-osd-0-86bf8cdc8-4nb5t` needs to be replaced and `worker-0` 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 will be *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 *terminating* state, use the `force` option to delete the pod.

```
oc delete pod rook-ceph-osd-0-86bf8cdc8-4nb5t --force --grace-period=0
```

Example output:

```
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may still be terminating.
pod "rook-ceph-osd-0-86bf8cdc8-4nb5t" force deleted
```

4. Remove the old OSD from the cluster so that you can add a new OSD.

- Identify the `DeviceSet` associated with the OSD to be replaced.

```
oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} | grep ceph.rook.io/pvc
```

Example output:

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-64xjl
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-64xjl
```

In this example, the Persistent Volume Claim (PVC) name is `ocs-deviceset-localblock-0-data-0-64xjl`.

- Identify the Persistent Volume (PV) associated with the PVC.

```
oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
```

where, `x`, `y`, and `pvc-suffix` are the values in the `DeviceSet` identified in an earlier step.

Example output:

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
localblock	ocs-deviceset-localblock-0-data-0-64xjl	Bound	local-pv-8137c873	256Gi	RWO	24h

In this example, the associated PV is `local-pv-8137c873`.

- Identify the name of the device to be replaced.

```
oc get pv local-pv-<pv-suffix> -o yaml | grep path
```

where, `pv-suffix` is the value in the PV name identified in an earlier step.

Example output:

```
path: /mnt/local-storage/localblock/vdc
```

In this example, the device name is `vdc`.

- Identify the `prepare-pod` associated with the OSD to be replaced.

```
oc describe -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix> | grep Used
```

where, `x`, `y`, and `pvc-suffix` are the values in the `DeviceSet` identified in an earlier step.

Example output:

```
Used By:    rook-ceph-osd-prepare-ocs-deviceset-localblock-0-data-0-64knzkc
```

- Delete any old `ocs-osd-removal` jobs.

```
$ oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

- Change to the `openshift-storage` project.

```
$ oc project openshift-storage
```

- Remove the old OSD from the cluster.

```
$ oc process -n openshift-storage ocs-osd-removal -p FAILED OSD IDS=${osd_id_to_remove} 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.

- 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
```

- 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
```

Important: If the `ocs-osd-removal-job` 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, where `<node name>` is the name of the node.

```
oc debug node/<node name>  
$ chroot /host
```

ii. Find a relevant device name based on the PVC names identified in the previous step, where `<pvc name>` is the name of the PVC.

```
dmsetup ls| grep <pvc name>
```

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 above command gets stuck due to insufficient privileges, run the following commands:

1. Press `CTRL+Z` to exit the above 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. Find the PV that need to be deleted.

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

local-pv-d6bf175b	1490Gi	RWO	Delete	Released	openshift-storage/ocs-deviceset-
0-data-0-6c5pw	localblock	2d22h	compute-1		

9. Delete the PV, where `<pv-name>` is name of the PV.

```
oc delete pv <pv-name>
```

10. Replace the old device and use the new device to create a new OpenShift Container Platform PV.

a. Log in to the OpenShift Container Platform node with the device to be replaced.

In this example, the OpenShift Container Platform node is `worker-0`.

```
$ oc debug node/worker-0
```

Example output:

```
Starting pod/worker-0-debug ...  
To use host binaries, run `chroot /host`  
Pod IP: 192.168.88.21  
If you don't see a command prompt, try pressing enter.  
# chroot /host
```

b. Record the `/dev/disk` that is to be replaced using the device name, `vdc`, identified earlier.

```
# ls -ahl /mnt/local-storage/localblock
```

Example output:

```
total 0  
drwxr-xr-x. 2 root root 17 Nov 18 15:23 .  
drwxr-xr-x. 3 root root 24 Nov 18 15:23 ..  
lrwxrwxrwx. 1 root root 8 Nov 18 15:23 vdc -> /dev/vdc
```

c. Find the name of the `LocalVolume` CR, and remove or comment out the device `/dev/disk` that is to be replaced.

```
$ oc get -n openshift-local-storage localvolume
```

Example output:

NAME	AGE
localblock	25h

```
# oc edit -n openshift-local-storage localvolume localblock
```

Example output:

```
[...]
  storageClassDevices:
    - devicePaths:
      #   - /dev/vdc
        storageClassName: localblock
        volumeMode: Block
[...]
```

Make sure to save the changes after editing the CR.

11. Log in to the OpenShift Container Platform node with the device to be replaced and remove the old **symlink**.

```
$ oc debug node/worker-0
```

Example output:

```
Starting pod/worker-0-debug ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.88.21
If you don't see a command prompt, try pressing enter.
# chroot /host
```

- a. Identify the old **symlink** for the device name to be replaced.

In this example, the device name is **vdc**.

```
# ls -alh /mnt/local-storage/localblock
```

Example output:

```
total 0
drwxr-xr-x. 2 root root 17 Nov 18 15:23 .
drwxr-xr-x. 3 root root 24 Nov 18 15:23 ..
lrwxrwxrwx. 1 root root 8 Nov 18 15:23 vdc -> /dev/vdc
```

- b. Remove the **symlink**.

```
# rm /mnt/local-storage/localblock/vdc
```

- c. Verify that the **symlink** is removed.

```
# ls -alh /mnt/local-storage/localblock
```

Example output:

```
total 0
drwxr-xr-x. 2 root root 6 Nov 18 17:11 .
drwxr-xr-x. 3 root root 24 Nov 18 15:23 ..
```

12. Replace the old device with the new device.

13. Log back into the correct OpenShift Container Platform node and identify the device name for the new drive.

The device name must change unless you are resetting the same device.

```
# lsblk
```

Example output:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
vda	252:0	0	40G	0	disk	
--vda1	252:1	0	4M	0	part	
--vda2	252:2	0	384M	0	part	/boot
`--vda4	252:4	0	39.6G	0	part	
--coreos-luks-root-nocrypt	253:0	0	39.6G	0	dm	/sysroot
vdb	252:16	0	512B	1	disk	
vdd	252:32	0	256G	0	disk	

In this example, the new device name is **vdd**.

14. After the new **/dev/disk** is available, you can add a new disk entry to the LocalVolume CR.

In this example, the new device is **/dev/vdd**.

```
# oc edit -n openshift-local-storage localvolume localblock
```

Example output:

```
[...]
  storageClassDevices:
    - devicePaths:
      #   - /dev/vdc
        - /dev/vdd
        storageClassName: localblock
        volumeMode: Block
[...]
```

Make sure to save the changes after editing the CR.

15. Verify that there is a new PV in *Available* state and of the correct size.

```
$ oc get pv | grep 256Gi
```

Example output:

local-pv-1e31f771	256Gi	RWO	Delete	Bound	openshift-storage/ocs-deviceset-localblock-2-data-0-6xhkf	localblock
24h						
local-pv-ec7f2b80	256Gi	RWO	Delete	Bound	openshift-storage/ocs-deviceset-localblock-1-data-0-hr2fx	localblock
24h						
local-pv-8137c873	256Gi	RWO	Delete	Available		

16. Create the new OSD for the new device.
Deploy the new OSD.

Restart the `rook-ceph-operator` to force operator reconciliation.
a. Identify the name of the `rook-ceph-operator`.

```
$ oc get -n openshift-storage pod -l app=rook-ceph-operator
```

Example output:

NAME	READY	STATUS	RESTARTS	AGE
rook-ceph-operator-85f6494db4-sg62v	1/1	Running	0	1d20h

b. Delete the `rook-ceph-operator`.

```
$ oc delete -n openshift-storage pod rook-ceph-operator-85f6494db4-sg62v
```

Example output:

```
pod "rook-ceph-operator-85f6494db4-sg62v" deleted
```

In this example, the `rook-ceph-operator` pod name is `rook-ceph-operator-85f6494db4-sg62v`.
c. Verify that the `rook-ceph-operator` pod is restarted.

```
$ oc get -n openshift-storage pod -l app=rook-ceph-operator
```

Example output:

NAME	READY	STATUS	RESTARTS	AGE
rook-ceph-operator-85f6494db4-wx9xx	1/1	Running	0	50s

Creation of the new OSD may take several minutes after the operator restarts.

17. Delete the `ocs-osd-removal` jobs.

```
$ 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 in a *Running* state.

```
oc get -n openshift-storage pods -l app=rook-ceph-osd
```

Example output:

rook-ceph-osd-0-76d8fb97f9-mn8qz	1/1	Running	0	23m
rook-ceph-osd-1-7c99657cfb-jdzvz	1/1	Running	1	25h
rook-ceph-osd-2-5f9f6dfb5b-2mnw9	1/1	Running	0	25h

2. Verify that a new PVC created.

```
oc get -n openshift-storage pvc | grep localblock
```

Example output:

ocs-deviceset-localblock-0-data-0-q4q6b	Bound	local-pv-8137c873	256Gi	RWO	localblock	10m
ocs-deviceset-localblock-1-data-0-hr2fx	Bound	local-pv-ec7f2b80	256Gi	RWO	localblock	1d20h
ocs-deviceset-localblock-2-data-0-6xhkf	Bound	local-pv-1e31f771	256Gi	RWO	localblock	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 previously identified nodes, 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.

Note: A full data recovery may take longer depending on the volume of data being recovered.

Replacing operational or failed storage devices on IBM Z or LinuxONE infrastructure

Use this information to replace operational or failed storage devices on IBM Z or LinuxONE infrastructure.

Before you begin

- IBM recommends that replacement devices are configured with similar infrastructure and resources to the device being replaced.
- 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-storesystem**.
- 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

You can replace operational or failed storage devices on IBM Z or LinuxONE infrastructure with new Small Computer System Interface (SCSI) disks.

IBM Z or LinuxONE supports SCSI FCP disk logical units (SCSI disks) as persistent storage devices from external disk storage. You can identify a SCSI disk using its FCP Device number, two target worldwide port names (WWPN1 and WWPN2), and the logical unit number (LUN). For more information, see Planning and Administration > z/VM: CP Planning and Administration > Storage Planning and Administration > Defining and Managing SCSI FCP Disks within [IBM z/VM documentation](#).

Procedure

1. List all the disks.

```
$ lszdev
```

Example output:

TYPE	ID	ON	PERS	NAMES
zfcp-host	0.0.8204	yes	yes	
zfcp-lun	0.0.8204:0x102107630b1b5060:0x4001402900000000	yes	no	sda sg0
zfcp-lun	0.0.8204:0x500407630c0b50a4:0x3002b0300000000	yes	yes	sdb sg1
qeth	0.0.bdd0:0.0.bdd1:0.0.bdd2	yes	no	encbddd0
generic-ccw	0.0.0009	yes	no	

A SCSI disk is represented as a **zfcp-lun** with the structure `<device-id>:<wwpn>:<lun-id>` in the **ID** section. The first disk is used for the operating system. If one storage device fails, you can replace it with a new disk.

2. Remove the disk.

Run the following command on the disk, replacing `scsi-id` with the SCSI disk identifier of the disk to be replaced:

```
chzdev -d scsi-id
```

For example, the following command removes one disk with the device ID **0.0.8204**, the WWPN **0x500507630a0b50a4**, and the LUN **0x4002403000000000**:

```
chzdev -d 0.0.8204:0x500407630c0b50a4:0x3002b0300000000
```

3. Append a new SCSI disk.

```
chzdev -e 0.0.8204:0x500507630b1b50a4:0x4001302a00000000
```

Note: The device ID for the new disk must be the same as the disk to be replaced. The new disk is identified with its WWPN and LUN ID.

4. List all the FCP devices to verify the new disk is configured.

```
lszdev zfcp-lun
```

Example output:

TYPE	ID	ON	PERS	NAMES
zfcp-lun	0.0.8204:0x102107630b1b5060:0x4001402900000000	yes	no	sda sg0
zfcp-lun	0.0.8204:0x500507630b1b50a4:0x4001302a00000000	yes	yes	sdb sg1

Monitoring Fusion Data Foundation

Learn how to monitor IBM Storage Fusion Data Foundation using the Block and File, and Object dashboards.

- **Cluster health**

Learn how to monitor the cluster health of IBM Storage 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.

Cluster health

Learn how to monitor the cluster health of IBM Storage 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.

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 will be 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](#).

Multicloud 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 multicloud dashboard on the Hub cluster.

- [Enabling multicloud dashboard on Hub cluster](#)

You can enable the multicloud dashboard on the install screen either before or after installing ODF Multicloud Orchestrator with the console plugin.

- [Verifying multicloud storage health on hub cluster](#)

Follow this procedure to verify multicloud storage health on hub cluster.

Enabling multicloud dashboard on Hub cluster

You can enable the multicloud dashboard on the install screen either before or after installing ODF Multicloud Orchestrator with the console plugin.

Before you begin

- Ensure that you have installed OpenShift Container Platform version 4.15 and have administrator privileges.
- Ensure that you have installed Red Hat Advanced Cluster Management (RHACM) for Kubernetes 2.9 from Operator Hub. For instructions on how to install, see [Installing RHACM](#) within the [Red Hat Advanced Cluster Management for Kubernetes](#) product documentation.
- Ensure you have enabled observability on RHACM. See [Enabling observability guidelines](#) within the [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 [Adding custom metrics](#) within the [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 will start 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 multicloud health, see [Verifying Fusion Data Foundation is healthy](#).

Verifying multicloud storage health on hub cluster

Follow this procedure to verify multicloud storage health on hub cluster.

Before you begin

Ensure that you have enabled multicloud monitoring, as described in [Enabling multicloud dashboard on Hub cluster](#).

Procedure

1. In the OpenShift web console of Hub cluster, ensure All Clusters is selected.
2. Navigate 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](#)
- [Enabling metadata on RBD and CephFS volumes](#)

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 will be 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 will not be 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.

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.

Consumer types	Metrics	Chart display
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

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

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 will not contain the metadata. Also, when you enable the metadata feature, the PVCs that were created before enabling will 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"\n' configmap/rook-ceph-operator-config patched
```

2. Wait for the respective CSI CephFS plugin provisioner pods and CSI RBD plugin 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
```

csi-cephfsplugin-b8d6c	2/2	Running	0	56m
csi-cephfsplugin-bnbg9	2/2	Running	0	56m
csi-cephfsplugin-kqd4	2/2	Running	0	56m
csi-cephfsplugin-provisioner-7dc78bb9b-q6dxb	5/5	Running	0	56m
csi-cephfsplugin-provisioner-7dc78bb9b-zc4q5	5/5	Running	0	56m
csi-rbdplugin-776dl	3/3	Running	0	56m
csi-rbdplugin-ff152	3/3	Running	0	56m
csi-rbdplugin-jx9mz	3/3	Running	0	56m
csi-rbdplugin-provisioner-5f6d766b6c-694fx	6/6	Running	0	56m
csi-rbdplugin-provisioner-5f6d766b6c-vzz45	6/6	Running	0	56m

- [Verify the metadata for RBD PVC](#)
 - [Verify the metadata for RBD clone](#)
 - [Verify the metadata for RBD snapshots](#)
 - [Verify the metadata for RBD Restore](#)
 - [Verify the metadata for CephFS PVC](#)
 - [Verify the metadata for CephFS clone](#)
 - [Verify the metadata for CephFS volume snapshot](#)
 - [Verify the metadata of CephFS Restore](#)
-

Verify the metadata for RBD PVC

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
```

	Bound	pvc-30628fa8-2966-499c-832d-a6a3a8ebc594	1Gi	RWO	ocs-
rbd-pvc					
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/namespace	openshift-storage

Verify the metadata for RBD 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

```

- Check the status of the clone.

```
oc get pvc | grep rbd-pvc
```

		Bound	pvc-30628fa8-2966-499c-832d-a6a3a8ebc594	1Gi	RWO	ocs-
rbd-pvc	storagecluster-ceph-rbd	15m				
rbd-pvc-clone	storagecluster-ceph-rbd	52s	pvc-0d72afda-f433-4d46-a7f1-a5fc3d766e0	1Gi	RWO	ocs-

- 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-a5fc3d766e0
csi.storage.k8s.io/pvc/name	rbd-pvc-clone
csi.storage.k8s.io/pvc/namespace	openshift-storage

Verify the metadata for RBD snapshots

Procedure

- 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

```

- 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	

- Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
rbd ls ocs-storagecluster-cephblockpool
```

```

csi-snap-ale24408-2848-11ed-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-ale24408-2848-11ed-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

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-
rbd-pvc	51m	Bound	pvc-30628fa8-2966-499c-832d-a6a3a8ebc594	1Gi	RWO	ocs-
storagecluster-ceph-rbd	47m	Bound	pvc-0d72afda-f433-4d46-a7f1-a5fc3d766e0	1Gi	RWO	ocs-
rbd-pvc-clone		Bound	pvc-f900e19b-3924-485c-bb47-01b84c559034	1Gi	RWO	ocs-
rbd-pvc-restore	32m	Bound				
storagecluster-ceph-rbd	111s	Bound				

3. Verify the metadata in the IBM Storage Ceph command-line interface (CLI).

```
rbd ls ocs-storagecluster-cephblockpool
```

```
csi-snap-ale24408-2848-11ed-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

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

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

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	CREATIONTIME	AGE	RESTORESIZE	SNAPSHOTCLASS
cephfs-pvc-snapshot	true	cephfs-pvc	cephfs-pvc-snapcontent-f0f17463-d13b-4e13-b44e-6340bbb3bee0	9s	9s	1Gi	ocs-storagecluster-cephfsplugin-snapclass

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

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	storagecluster-cephfs	Bound	pvc-4151128c-86f0-468b-b6e7-5fdfb51ba1b9	1Gi	RWO	ocs-
	cephfs-pvc-clone	29m	pvc-3d4c4e78-f7d5-456a-aa6e-4da4a05ca4ce	1Gi	RWX	ocs-
	storagecluster-cephfs	20m				
	cephfs-pvc-restore		pvc-43d55ea1-95c0-42c8-8616-4ee70b504445	1Gi	RWX	ocs-
	storagecluster-cephfs	21s				

- 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-43d55eal-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*.

Troubleshooting

Administrators can use this troubleshooting information to understand how to troubleshoot and fix their IBM Storage 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.

- [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 Storage 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](#)

- [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](#)

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/ocs-must-gather-rhel8:v4.12 <local-registry>/odf4/ocs-must-gather-rhel8:v4.12 [--registry-config=<path-to-the-registry-config>] [--insecure=true]

<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 Storage 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 Storage 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 Troubleshooting within [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/ocs-must-gather-rhel8:v4.12 --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/ocs-must-gather-rhel8:v4.12 --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/ocs-must-gather-rhel8:v4.12 --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/ocs-must-gather-rhel8:v4.12 --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/ocs-must-gather-rhel8:v4.12 --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).

## 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 csi-cephfsplugin-<ID> -n openshift-storage -c csi-cephfsplugin
```

```
oc logs csi-rbdplugin-<ID> -n openshift-storage -c csi-rbdplugin
```

- To generate logs for all the containers in the CSI pod:

```
oc logs csi-cephfsplugin-<ID> -n openshift-storage --all-containers
```

```
oc logs csi-rbdplugin-<ID> -n openshift-storage --all-containers
```

- Generating logs for cephfs or rbd provisioner pods to detect problems if PVC is not in *BOUND* state:

```
oc logs csi-cephfsplugin-provisioner-<ID> -n openshift-storage -c csi-cephfsplugin
```

```
oc logs csi-rbdplugin-provisioner-<ID> -n openshift-storage -c csi-rbdplugin
```

- To generate logs for all the containers in the CSI pod:

```
oc logs csi-cephfsplugin-provisioner-<ID> -n openshift-storage --all-containers
```

```
oc logs csi-rbdplugin-provisioner-<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
```

```
crlctl images | grep <component>
```

For Example :

```
crlctl 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 level of logs

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.

In order 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

- Specify a blank node selector for the openshift-storage namespace.

```
oc annotate namespace openshift-storage openshift.io/node-selector=
```

- Delete the original pods generated by the DaemonSets.

```
oc delete pod -l app=csi-cephfsplugin -n openshift-storage
oc delete pod -l app=csi-rbdplugin -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 Storage 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 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.

- **[Troubleshooting unhealthy blocklisted nodes](#)**

## Resolving alerts and errors

IBM Storage 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 > Cluster](#)
- [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 is 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 is 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 is in progress, this alert is temporary.
  - If an update is not in progress, restart the upgrade process.
2. Once the upgrade is complete, check for alerts and operator status.
3. If the issue persists 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](#).

---

## 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

In order 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 will become read-only at 85%. Your Ceph cluster will become 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 will become read-only at 85%. Free up some space or expand the storage cluster.
- **CephClusterReadOnly**  
Storage cluster utilization has crossed 85% and will become read-only now. 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 will 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.
- **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 will 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 will 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 will 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.
- **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.
- **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.

## CephClusterCriticallyFull

Storage cluster utilization has crossed 80% and will become read-only at 85%. Your Ceph cluster will become 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 the [Scaling storage guide](#).

## 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 this impacts 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 will become 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 the [Scaling storage guide](#).

## 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 will become read-only now. 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 the [Scaling storage guide](#).

## 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 will continue to function in this state, it is recommended to fix the errors so that the cluster does not get into an error state. 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
```

---

## 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 running impacts 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 will 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 will 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 will 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 Storage 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 will resolve the alert through self healing processes.

Important: This is only applicable to IBM Storage 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 via the add-on and the cluster will resolve 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

- 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`
  - 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>`
  - 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

- 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'
```

- 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

- 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`
- 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>`
- 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 the [Scaling storage guide](#).

---

## 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-z1qkx`, <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.

- 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](#).

## 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 will disable 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 will disable the quota.

## 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 will disable 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 will disable the quota.

---

## 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 on 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 on 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 on 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 on 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 *ReadWriteOnce* (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
```

## Troubleshooting unhealthy blocklisted nodes

### ODFRBDCClientBlocked

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 <code>ocs_rbd_client_blocklisted</code> metric reports a value of 1 for the node. Additionally, there are pods in a <code>CreateContainerError</code> 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. Once 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 Storage 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 |
|----------------|--------|------------------------------------------|----------|--------------|--------------|
| db-noobaa-db-0 | Bound  | pvc-d96c747b-2ab5-47e2-b07e-1079623748d8 | 50Gi     | RWO          | ocs-         |

|                                      |                    |                                          |  |                     |                  |                         |  |
|--------------------------------------|--------------------|------------------------------------------|--|---------------------|------------------|-------------------------|--|
| <code>storagecluster-ceph-rbd</code> | <code>114s</code>  |                                          |  |                     |                  |                         |  |
| <code>ocs-deviceset-0-0-1zfrd</code> | <code>Bound</code> | <code>local-pv-7e70c77c<br/>2m10s</code> |  | <code>1769Gi</code> | <code>RWO</code> | <code>localblock</code> |  |
| <code>ocs-deviceset-1-0-7rggl</code> | <code>Bound</code> | <code>local-pv-b19b3d48<br/>2m10s</code> |  | <code>1769Gi</code> | <code>RWO</code> | <code>localblock</code> |  |
| <code>ocs-deviceset-2-0-znhk8</code> | <code>Bound</code> | <code>local-pv-e9f22cdc<br/>2m10s</code> |  | <code>1769Gi</code> | <code>RWO</code> | <code>localblock</code> |  |

## Removing failed or unwanted Ceph Object Storage devices

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](#).
- [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](#)
- [Troubleshooting the error cephosd:osd.0 is NOT ok to destroy while removing failed or unwanted Ceph OSDs](#)

## Verifying Ceph cluster is healthy

### 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

### 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 Storage 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

---

### 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.

- 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
```

- Get the persistent volume (PV) associated with the PVC.

```
oc get -n openshift-storage pvc <pvc-name>
```

- Get the failed device name.

```
oc get pv <pv-name-from-above-command> -oyaml | grep path
```

- Get the prepare-pod associated with the failed OSD.

```
oc describe -n openshift-storage pvc ocs-deviceset-0-0-nvs68 | grep Mounted
```

- Delete the osd-prepare pod before removing the associated PVC.

```
oc delete -n openshift-storage pod <osd-pod-name>
```

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).

- Log in to node with the failed device.

```
oc debug node/<node-with-failed-osd>
```

- Record the `/dev/disk/by-id/<id>` for the failed device name.

```
ls -ahl /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>
```

- Get the OSD symlink for the failed device name.

```
ls -ahl /mnt/local-storage/localblock
```

- 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 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

---

- 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
```

- 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:

- Get the object kind of the resource which needs to be removed.

See the message in the above output.

Example, where `cephobjectstoreuser.ceph.rook.io` is the object kind.

```
message: Some content in the namespace has finalizers remaining:
cephobjectstoreuser.ceph.rook.io
```

- 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 |

- 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

- Check for CephFS pvc stuck in *Pending* status.

```
oc get pvc -n <namespace>
```

Example output :

| NAME                  | STATUS       | VOLUME | AGE |
|-----------------------|--------------|--------|-----|
| CAPACITY ACCESS MODES | STORAGECLASS |        |     |
| 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 will 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 1Mi RWO | Bound pvc-1ac0c6e6-9428-445d-bbd6-1284d54ddb47 | 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 -o name);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 -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 -o jsonpath='{ .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 -o name);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 -o name | 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 -o name)
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 -o name)
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 = AQDxTF1hpzgu0xAA0sS8nN4udoO350Eb3bqMQ==
caps mds = "allow "
caps mgr = "allow *"
caps mon = "allow *"
caps osd = "allow *"
[mds.ocs-storagecluster-cephfilesystem-a]
key = AQCKTV1horgjARAA8aF/BDh/4+eG4RCNBC1+aw==
caps mds = "allow"
caps mon = "allow profile mds"
caps osd = "allow *"
[mds.ocs-storagecluster-cephfilesystem-b]
key = AQCKTV1hN4gKLBA5emIVq3ncV7AMEM1c1RmGA==
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+eIZLZtpfkchvTRg==
caps mds = "allow *"
caps mon = "allow profile mgr"
caps osd = "allow *"
[client.crash]
key = AQBOTV1htO1aGRAe2MPYcGdiAT+Oo4CNPSF1g==
caps mgr = "allow rw"
caps mon = "allow profile crash"
[client.csi-cephfs-node]
key = AQBOTV1hiAtuBAAaPPBVgh1AqZJlDeHWdoFLw==
caps mds = "allow rw"
caps mgr = "allow rw"
caps mon = "allow r"
caps osd = "allow rw tag cephfs *="
[client.csi-cephfs-provisioner]
key = AQBNTV1hHu6wBAAAzNXZv36aZJuE1iz7S7GfeQ==
caps mgr = "allow rw"
caps mon = "allow r"
caps osd = "allow rw tag cephfs metadata="
[client.csi-rbd-node]
key = AQBNTV1hLnkIRAAWnpIN9bUAmSHoVJ0EJXHRw==
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 -o name)
```

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 -o yaml | 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> -o yaml | 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> -o yaml)
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

```

Important: If the filesystem is offline or MDS service is missing, you need to restore the CephFS. For more information, see [Restoring the CephFS](#).  
 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-mon**s 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: Make sure 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` 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 \
 --extract-monmap ${monmap_path}
```

```
--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: 2
 memory: 8Gi
 requests:
 cpu: 2
 memory: 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` variable 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

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
```

## Configuring Data Foundation for Disaster Recovery

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.

The Fusion Data Foundation Disaster Recovery (DR) capability enables DR across multiple Red Hat OpenShift Container Platform clusters, and is categorized as follows:

- Metro-DR**

Metro-DR ensures business continuity during the unavailability of a data center with no data loss. In the public cloud these would be similar to protecting from an Availability Zone failure.

- Regional-DR**

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 this would be similar to protecting from a region failure.

- **Disaster Recovery with stretch cluster**

Stretch cluster solution ensures business continuity with no-data loss disaster recovery protection with OpenShift Data Foundation based synchronous replication in a single OpenShift cluster, stretched across two data centers with low latency and one arbiter node.

Zone failure in Metro-DR and region failure in Regional-DR is usually expressed 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 will be lost or need 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?"

The intent of this guide is to detail the Disaster Recovery steps and commands necessary to be able to failover an application from one OpenShift Container Platform cluster to another and then relocate the same application to the original primary cluster.

## 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 Storage Fusion Data Foundation 4.14 Metro-DR or Regional-DR solution. For details, see [Metro-DR solution for Fusion Data Foundation](#) or [Regional-DR solution for Fusion Data Foundation](#)
2. Deploy IBM Storage Fusion 2.7, as detailed in [Deploying IBM Storage Fusion](#)

The IBM Metro- & Regional- Disaster Recovery instructions helps you to configure your infrastructure in order 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 solution for Fusion Data Foundation\*\*](#)

This section of the guide provides you with insights into the IBM Storage Fusion Data Foundation Metropolitan Disaster Recovery (Metro-DR) solution along with the steps and commands necessary to be able to failover an application from one OpenShift Container Platform cluster to another and then failback the same application to the original primary cluster.

- [\*\*Regional-DR solution for Fusion Data Foundation\*\*](#)

This section of the guide provides you with insights into the IBM Storage Fusion Data Foundation Regional Disaster Recovery (Regional-DR) solution along with the steps and commands necessary to be able to failover an application from one 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 Storage 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.

---

## Metro-DR solution for Fusion Data Foundation

This section of the guide provides you with insights into the IBM Storage Fusion Data Foundation Metropolitan Disaster Recovery (Metro-DR) solution along with the steps and commands necessary to be able to failover an application from one 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 easily 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.

- [\*\*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 Storage 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 will be 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 Storage 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](#)  
In order 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 OpenShift Container Platform's OperatorHub 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 Storage 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 fallback (relocate) to this new cluster.
  - [Hub recovery using Red Hat Advanced Cluster Management \[Technology preview\]](#)
- 

## 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 Storage 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 Storage Fusion Data Foundation DR Hub Operator

Automatically installed as part of IBM Storage Fusion Data Foundation Multicluster Orchestrator installation on the hub cluster to orchestrate failover or relocation of DR enabled applications.

#### IBM Storage 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.

---

## 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 Storage 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 will be 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](#).

---

## Requirements for enabling Metro-DR

This section lists all the prerequisites for implementing the Disaster Recovery solution supported by IBM Storage 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,
    - Navigate 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 Ceph's 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](#)

Once 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 fulfill 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 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 will be 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.

- 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"
}

```

- Navigate to the /usr/share/cephadm-ansible directory.
- 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 or [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 will be 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 **cephadm 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. Once the bootstrap finishes, you will see the following output 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.lqmqt 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 Req: 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.ogcqk1
```

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

Once 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
```

To know more 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
```

To know more 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 will not select this rule. In this example, `1`.

#### max\_size

If a pool makes more replicas than this number, CRUSH will 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 its 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 will increase (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

In order 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.e18cp
rgw.objectgw.ceph6.mecpzm ceph6 *:8080 running (5d) 31s ago 7w 204M - 16.2.7-112.e18cp
```

```
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-storesystem > 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 OpenShift Container Platform's OperatorHub on the Hub cluster.

## Procedure

1. Go to Hub cluster > OperatorHub and use the keyword filter to search for ODF Multicluster Orchestrator.
2. Click the IBM Storage Fusion Data Foundation 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 Storage 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 from 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 > Data policies.

2. 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 to which this new policy will be associated with.

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 not supported in DRClusters.

```
oc get drcluster <drcluster_name> -o jsonpath='{.status.conditions[2].reason}{"\n"}'
```

Example output: Succeeded

Note: Make sure to run command for both DRClusters on the Hub cluster.

9. Verify that the IBM Storage 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                                                                    | REPLACES | PHASE | DISPLAY                       | VERSION  |
|-------------------------------------------------------------------------|----------|-------|-------------------------------|----------|
|                                                                         |          |       | Openshift DR Cluster Operator | 4.14.0   |
| clusterserviceversion.operators.coreos.com/odr-cluster-operator.v4.14.0 |          |       |                               |          |
| Succeeded                                                               |          |       |                               |          |
| clusterserviceversion.operators.coreos.com/volsync-product.v0.8.0       |          |       | VolSync                       | 0.8.0    |
| Succeeded                                                               |          |       |                               |          |
| NAME                                                                    |          | READY | STATUS                        | RESTARTS |
| pod/ramen-dr-cluster-operator-6467cf5d4c-cc8kz                          |          | 2/2   | Running                       | 0        |
| AGE                                                                     |          |       |                               |          |
| 3d12h                                                                   |          |       |                               |          |

You can also verify that IBM Storage 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.

## 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
```

Once 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. Make sure to add these annotations for both DRCluster resources (for example: `ocp4perf1` and `ocp4perf2`).

---

## Create sample application for testing disaster recovery solution

IBM Storage Fusion Data Foundation disaster recovery (DR) solution supports disaster recovery for applications that are managed by Red Hat Advanced Cluster Management for Kubernetes (RHACM).

For more information, see Applications > Managing Applications within [Red Hat Advanced Cluster Management for Kubernetes](#) product documentation.

The following sections detail how to create an application and apply a DRPolicy to an application.

- [Subscription-based applications](#)  
OpenShift users that do not have cluster-admin permissions, see the [knowledge article](#) on how to assign necessary permissions to an application user for executing disaster recovery actions.
- [ApplicationSet-based applications](#)  
OpenShift users that do not have cluster-admin permissions cannot create ApplicationSet-based applications.
- [Deleting sample application](#)  
The sample application `busybox` can be deleted by using the Red Hat Advanced Cluster Management for Kubernetes (RHACM) console.

---

## Creating subscription-based sample application

To test `failover` from the Primary-managed cluster to the Secondary-managed cluster and `relocate`, use the following sample application.

### Before you begin

---

- Ensure that the Red Hat OpenShift GitOps operator is installed on the Hub cluster. For instructions, see [RHACM documentation](#).
- When creating an application for general consumption, ensure that the application is deployed to **only** one cluster.
- Use the sample application called `busybox` as an example.
- Ensure that all external routes of the application are configured by using either Global Traffic Manager (GTM) or Global Server Load Balancing (GLSB) service, for traffic redirection when the application fails over or is relocated.
- As a best practice, group Red Hat Advanced Cluster Management (RHACM) subscriptions that belong together to refer to a single Placement Rule to DR protect them as a group. Further create them as a single application for a logical grouping of the subscriptions for future DR actions like failover and relocate.  
Note: If unrelated Subscriptions point to the same PlacementRule for placement actions, they will also be DR protected as the DR workflow controls all Subscriptions that point to a PlacementRule.

## Procedure

---

1. On the Hub cluster, navigate to Applications > Create application .
2. Select type as Subscription.
3. Enter your application Name (for example, `busybox`) and Namespace (for example, `busybox-sample`).
4. In the Repository location for resources section, select Repository type Git.
5. Enter the Git repository URL for the sample application, the GitHub Branch, and Path where the resources `busybox` Pod and PVC will be created.  
Use the sample application repository as <https://github.com/red-hat-storage/ocm-ramen-samples>, where the Branch is release-4.14 and Path is busybox-odr-metro.
6. Scroll down in the form until you see Deploy application resources on clusters with all specified labels.
  - Select the global Cluster sets or the one that includes the correct managed clusters for your environment.
  - Add a label `<name>` with its value set to the managed cluster name.
7. Click Create which is at the upper right hand corner.  
On the follow-on screen go to the Topology tab. There you can see all green checkmarks on the application topology.  
Note: To get more information, click any of the topology elements and a window will appear on the right of the topology view.
8. Validate the sample application deployment.

After the **busybox** application has been deployed to your preferred Cluster, the deployment can be validated.

Login to your managed cluster where **busybox** was deployed by using RHACM.

```
oc get pods,pvc -n busybox-sample
```

Example output:

| NAME                        | READY | STATUS  | RESTARTS | AGE |
|-----------------------------|-------|---------|----------|-----|
| pod/busybox-67bf494b9-zl5tr | 1/1   | Running | 0        | 77s |

| NAME                              | STORAGECLASS            | AGE | STATUS | VOLUME                                    | CAPACITY | ACCESS MODES |
|-----------------------------------|-------------------------|-----|--------|-------------------------------------------|----------|--------------|
| persistentvolumeclaim/busybox-pvc | storagecluster-ceph-rbd | 77s | Bound  | pvc-c732e5fe-daaaf-4c4d-99dd-462e04c18412 | 5Gi      | RWO ocs-     |

## Apply DRPolicy to sample application

### Before you begin

Ensure that both managed clusters referenced in the DRPolicy are reachable. If not, the application will not be 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. You can also use the Add application resource option to add multiple 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.
  - c. Click View more details to view the status of ongoing activities with the policy in use with the application.

## Creating ApplicationSet-based applications

### Before you begin

- Ensure that the Red Hat OpenShift GitOps operator is installed on the Hub cluster. For instructions, see [RHACM documentation](#).
- Ensure that both Primary and Secondary managed clusters are registered to GitOps. For registration instructions, see [Registering managed clusters to GitOps](#). 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 [Configuring application placement tolerations for Red Hat Advanced Cluster Management and OpenShift GitOps](#).

### Procedure

1. On the Hub cluster, navigate to All Clusters > Applications > Create application.
2. Select type as Argo CD ApplicationSet - Push model.
3. In General step 1, 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 will be created.
  - a. Use the sample application repository as <https://github.com/red-hat-storage/ocm-ramen-samples>
  - b. Select Revision as release-4.15
  - 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

### Before you begin

Ensure that both managed clusters referenced in the DRPolicy are reachable. If not, the application will not be 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 sample application

The sample application **busybox** can be deleted by using the Red Hat Advanced Cluster Management for Kubernetes (RHACM) console.

### About this task

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.

### 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. Navigate 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:

Once the managed cluster is fenced, all communication from applications to the Fusion Data Foundation external storage cluster fails and some Pods will be 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 will failover 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. Navigate 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:
- Once the managed cluster is fenced, all communication from applications to the Fusion Data Foundation external storage cluster fails and some Pods will be 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. Navigate 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: Make sure 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. Navigate 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: Make sure 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	READY	STATUS	RESTARTS	AGE	NAME
-----------	-------	--------	----------	-----	------

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 fallback (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**.

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 will show 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 will not be 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 multicloud.operator.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
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.

- Detach the replacement cluster.
- Create a new OpenShift cluster (recovery cluster) and import the new cluster into the RHACM console. For instructions, see [Creating a cluster](#) and [Importing a target managed cluster to the hub cluster](#).

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 greater) and the same version is on the surviving cluster.

11. On the hub cluster, install the ODF Multicloud 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.

- Select Create DRPolicy and name your policy.
- Select the recovery cluster and the surviving cluster.
- 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]

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](#)
- [Switching to passive hub cluster](#)

Configuring passive hub cluster

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 multicloud 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 Multicloud 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 `VolumeSynchronizationDelay` alert using this command. For alert details, see [Disaster recovery alerts](#) chapter.

Switching to passive hub cluster

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 [Restoring a hub cluster](#) from backup. Important: Recovering a failed hub to its passive instance will only restore 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. Verify that the Primary and Secondary 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 will not be successfully imported.
3. Wait until DRPolicy validation succeeds.
4. 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

5. Refresh the RHACM console to make the DR monitoring dashboard tab accessible if it was enabled on the Active hub cluster.
6. 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.
7. 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](#).
8. 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 solution for Fusion Data Foundation

This section of the guide provides you with insights into the IBM Storage Fusion Data Foundation Regional Disaster Recovery (Regional-DR) solution along with the steps and commands necessary to be able to failover an application from one OpenShift Container Platform cluster to another and then failback the same application to the original primary cluster.

- [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.

- [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 Storage Fusion Data Foundation across two distinct OpenShift Container Platform clusters. In addition to two managed clusters, a third OpenShift Container Platform cluster will be 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 Storage Fusion Data Foundation.
- [Creating a Fusion Data Foundation cluster on managed clusters](#)
- [Installing Fusion Data Foundation Multicluster Orchestrator operator](#)
Fusion Data Foundation Multicluster Orchestrator is a controller that is installed from OpenShift Container Platform's OperatorHub 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.
- [Create sample application for testing disaster recovery solution](#)
IBM Storage 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](#)
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. 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.
- [ApplicationSet-based application failover between managed clusters](#)
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. 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.
- [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.
- [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 \[Technology preview\]](#)

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 4.15 and Red Hat Advanced Cluster Management for Kubernetes 2.9 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 > 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 Storage 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 Storage Fusion Data Foundation DR Hub Operator

Automatically installed as part of IBM Storage Fusion Data Foundation Multicluster Orchestrator installation on the hub cluster to orchestrate failover or relocation of DR enabled applications.

IBM Storage 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.

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 Storage Fusion Data Foundation across two distinct OpenShift Container Platform clusters. In addition to two managed clusters, a third OpenShift Container Platform cluster will be 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 sample 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 Subscription-based application between managed clusters](#).
 - 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 between managed clusters](#) and [Relocating an ApplicationSet-based application between managed clusters](#).

Requirements for enabling Regional-DR

This section lists all the prerequisites for implementing the Disaster Recovery solution supported by IBM Storage 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 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](#). 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,
 - Navigate 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.

Note: Regional-DR is supported with Fusion Data Foundation 4.15 and Red Hat Advanced Cluster Management for Kubernetes 2.9 combinations only.

- Connect the private 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"
  ]
}
```

For more information, see [Submariner documentation](#).

Creating a Fusion Data Foundation cluster on managed clusters

About this task

In order 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 Fusion Data Foundation deployment guides and instructions that are specific to your infrastructure (AWS, VMware, BM, Azure, etc.).

Procedure

1. Install and configure the latest **Fusion Data Foundation** cluster on each of the managed clusters.

For information about the Fusion Data Foundation deployment, refer to your [infrastructure specific deployment guides](#) (for example, AWS, VMware, Bare metal, Azure).

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 above 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 OpenShift Container Platform's OperatorHub on the Hub cluster.

Procedure

1. Go to Hub cluster > OperatorHub and use the keyword filter to search for ODF Multicluster Orchestrator.
2. Click the IBM Storage Fusion Data Foundation 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 Storage 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 > Data policies.
2. 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 to which this new policy will be associated with.
5. Replication policy is automatically set to **Asynchronous** (async), based on the OpenShift clusters selected and a Sync schedule option will become 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. Click Create.
8. 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.

9. 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: Make sure to run command for both DRClusters on the Hub cluster.

10. Verify that the IBM Storage 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 Succeeded	Openshift DR Cluster Operator	4.14.0
clusterserviceversion.operators.coreos.com/volsync-product.v0.8.0 Succeeded	VolSync	0.8.0
NAME pod/ramen-dr-cluster-operator-6467cf5d4c-cc8kz	READY STATUS RESTARTS AGE	2/2 Running 0 3d12h

You can also verify that IBM Storage 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.

11. Verify that the status of the IBM Storage 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.

Create sample application for testing disaster recovery solution

IBM Storage Fusion Data Foundation disaster recovery (DR) solution supports disaster recovery for applications that are managed by Red Hat Advanced Cluster Management for Kubernetes (RHACM).

For more information, see Applications > Managing Applications within [Red Hat Advanced Cluster Management for Kubernetes](#) product documentation.

The following sections detail how to create an application and apply a DRPolicy to an application.

- [Subscription-based applications](#)

OpenShift users that do not have cluster-admin permissions, see the [knowledge article](#) on how to assign necessary permissions to an application user for executing disaster recovery actions.

- [ApplicationSet-based applications](#)

OpenShift users that do not have cluster-admin permissions cannot create ApplicationSet-based applications.

- [Deleting sample application](#)

The sample application **busybox** can be deleted by using the Red Hat Advanced Cluster Management for Kubernetes (RHACM) console.

Creating a sample subscription-based application

To test **failover** from the Primary-managed cluster to the Secondary-managed cluster and **relocate**, use the following sample application.

Before you begin

- Ensure that the Red Hat OpenShift GitOps operator is installed on the Hub cluster. For instructions, see [RHACM documentation](#).
- When creating an application for general consumption, ensure that the application is deployed to **only** one cluster.
- Use the sample application called **busybox** as an example.
- Ensure that all external routes of the application are configured by using either Global Traffic Manager (GTM) or Global Server Load Balancing (GLSB) service, for traffic redirection when the application fails over or is relocated.
- As a best practice, group Red Hat Advanced Cluster Management (RHACM) subscriptions that belong together to refer to a single Placement Rule to DR protect them as a group. Further create them as a single application for a logical grouping of the subscriptions for future DR actions like failover and relocate.
Note: If unrelated subscriptions refer to the same Placement Rule for placement actions, they are also DR protected as the DR workflow controls all subscriptions that references the Placement Rule.

Procedure

1. On the Hub cluster, navigate to Applications > Create application.
2. Select type as Subscription.
3. Enter your application Name (for example, **busybox**) and Namespace (for example, **busybox-sample**).
4. In the Repository location for resources section, select Repository type Git.
5. Enter the Git repository URL for the sample application, the GitHub Branch, and Path where the resources **busybox** Pod and PVC will be created.

- Use the sample application repository as <https://github.com/red-hat-storage/ocm-ramen-samples>,
 - Select Branch as release-4.14
 - Choose one of the following Path
 - busybox-odr to use RBD Regional-DR.
 - busybox-odr-cephfs to use CephFS Regional-DR.
6. Scroll down in the form until you see Deploy application resources on clusters with all specified labels.
- Select the global Cluster sets or the one that includes the correct managed clusters for your environment.
 - Add a label <name> with its value set to the managed cluster name.
7. Click Create which is at the upper right hand corner.
- On the follow-on screen go to the Topology tab. There you can see all green checkmarks on the application topology.
Note: To get more information, click any of the topology elements.
8. Validate the sample application deployment.

After the **busybox** application has been deployed to your preferred Cluster, the deployment can be validated.

Log in to your managed cluster where **busybox** was deployed by using RHACM.

```
oc get pods,pvc -n busybox-sample
```

Example output:

NAME	READY	STATUS	RESTARTS	AGE
pod/busybox-67bf494b9-zl5tr	1/1	Running	0	77s

NAME	STORAGECLASS	AGE	STATUS	VOLUME	CAPACITY	ACCESS MODES
persistentvolumeclaim/busybox-pvc	persistentvolumecluster-ceph-rbd	77s	Bound	pvc-c732e5fe-daaaf-4c4d-99dd-462e04c18412	5Gi	RWO ocs-

Apply DRPolicy to sample application

Before you begin

Ensure that both managed clusters referenced in the DRPolicy are reachable. If not, the application will not be 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. You can also use the Add application resource option to add multiple 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.
 - c. Click View more details to view the status of ongoing activities with the policy in use with the application.
9. Optional: Verify RADOS block device (RBD) **volumereplication** and **volumereplicationgroup** on the primary cluster.

```
oc get volumereplications.replication.storage.openshift.io
```

Example output:

NAME	AGE	VOLUMEREPLICATIONCLASS	PVCNAME	DESIREDSTATE	CURRENTSTATE
busybox-pvc	2d16h	rbd-volumereplicationclass-1625360775	busybox-pvc	primary	Primary

```
oc get volumereplicationgroups.ramendr.openshift.io
```

Example output:

NAME	DESIREDSTATE	CURRENTSTATE
busybox-drpc	primary	Primary

10. Optional: Verify CephFS volsync **replicationsource** has been set up successfully in the primary cluster and VolSync **ReplicationDestination** has been set up in the failover cluster.

```
oc get replicationsource -n busybox-sample
```

Example output:

NAME	SOURCE	LAST SYNC	DURATION	NEXT SYNC
busybox-pvc	busybox-pvc	2022-12-20T08:46:07Z	1m7.794661104s	2022-12-20T08:50:00Z

```
oc get replicationdestination -n busybox-sample
```

Example output:

NAME	LAST SYNC	DURATION	NEXT SYNC
busybox-pvc	2022-12-20T08:46:32Z	4m39.52261108s	

Creating ApplicationSet-based applications

Before you begin

- Ensure that the Red Hat OpenShift GitOps operator is installed on the Hub cluster. For instructions, see [RHACM documentation](#).
- Ensure that both Primary and Secondary managed clusters are registered to GitOps. For registration instructions, see [Registering managed clusters to GitOps](#). 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 [Configuring application placement tolerations for Red Hat Advanced Cluster Management and OpenShift GitOps](#).

Procedure

1. On the Hub cluster, navigate to All Clusters > Applications > Create application.
2. Select type as Argo CD ApplicationSet - Push model.
3. In General step 1, 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 will be created.
 - a. Use the sample application repository as <https://github.com/red-hat-storage/ocm-ramen-samples>
 - b. Select Revision as release-4.15
 - c. Choose one of the following Path:
 - busybox-odr to use RBD Regional-DR.
 - busybox-odr-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.

Apply Data policy to sample ApplicationSet-based application

Before you begin

Ensure that both managed clusters referenced in the DRPolicy are reachable. If not, the application will not be 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 **PVC 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. Optional: Verify RADOS block device (RBD) **volumereplication** and **volumereplicationgroup** on the primary cluster.

```
oc get volumereplications.replication.storage.openshift.io
```

Example output:

NAME	AGE	VOLUMEREPLICATIONCLASS	PVCNAME	DESIREDSTATE	CURRENTSTATE
busybox-pvc	2d16h	rbd-volumereplicationclass-1625360775	busybox-pvc	primary	Primary

```
oc get volumereplicationgroups.ramendr.openshift.io
```

Example output:

NAME	DESIREDSTATE	CURRENTSTATE
busybox-drpc	primary	Primary

10. Optional: Verify CephFS volsync `replicationsource` has been set up successfully in the primary cluster and VolSync `ReplicationDestination` has been set up in the failover cluster.

```
oc get replicationsource -n busybox-sample
```

Example output:

NAME	SOURCE	LAST SYNC	DURATION	NEXT SYNC
busybox-pvc	busybox-pvc	2022-12-20T08:46:07Z	1m7.794661104s	2022-12-20T08:50:00Z

```
oc get replicationdestination -n busybox-sample
```

Example output:

NAME	LAST SYNC	DURATION	NEXT SYNC
busybox-pvc	2022-12-20T08:46:32Z	4m39.52261108s	

Deleting sample application

The sample application `busybox` can be deleted by using the Red Hat Advanced Cluster Management for Kubernetes (RHACM) console.

About this task

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.

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 \rightarrow 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

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. 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.

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. Navigate to RHACM console \rightarrow Infrastructure \rightarrow Clusters \rightarrow 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"
```

Procedure

- On the Hub cluster, navigate to Applications.
 - Click the Actions menu at the end of application row to view the list of available actions.
 - Click Failover application.
 - After the **Failover application** popup is shown, select Policy and Target cluster to which the associated application will failover in a disaster.
 - 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.
- 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 7.
 - If the status is *Unknown* or *Not ready*, then wait until the status changes to *Ready*.
 - Click Initiate.

All the system workloads and their available resources are now transferred to the target cluster.
 - Close the modal window and track the status using the Data policy column on the Applications page.
 - Verify that the activity status shows as *FailedOver* for the application.
 - Navigate to the Applications > Overview tab.
 - In the Data policy column, click the policy link for the application you applied the policy to.
 - On the Data Policy popover page, click the View more details link.
 - 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

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. 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.

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. Navigate 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"
```

Procedure

- On the Hub cluster, navigate to Applications.
- Click the Actions menu at the end of application row to view the list of available actions.
- Click Failover application.
- 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.
- Click Initiate.

All the system workloads and their available resources are now transferred to the target cluster.
- Close the modal window and track the status using the Data policy column on the Applications page.
- Verify that the activity status shows as *FailedOver* for the application.
 1. Navigate to the Applications > Overview tab.
 2. In the Data policy column, click the policy link for the application you applied the policy to.
 3. 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 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. Navigate 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. On the Hub cluster, navigate to Applications.
2. Click the Actions menu at the end of application row to view the list of available actions.
3. Click Relocate application.
4. When the Relocate application modal is shown, select Policy and Target cluster to which the associated application migrates to.
5. 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.
6. 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 7.
 - If the status is *Unknown* or *Not ready*, then wait until the status changes to *Ready*.
7. Click Initiate.
All the system workloads and their available resources are now transferred to the target cluster.
8. Close the modal window and track the status using the Data policy column on the Applications page.
9. Verify that the activity status shows as *Relocated* for the application.
 - a. Navigate 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 \[Technology preview\]](#).
- 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. Navigate 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. On the Hub cluster, navigate to Applications.
2. Click the Actions menu at the end of application row to view the list of available actions.
3. Click Relocate application.
4. When the Relocate application modal is shown, select Policy and Target cluster to which the associated application will relocate to in case of a disaster.
5. Click Initiate.
All the system workloads and their available resources are now transferred to the target cluster.
6. Close the modal window and track the status using the Data policy column on the Applications page.
7. 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.

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 [Technology preview]

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](#)
- [Switching to passive hub cluster](#)

Configuring passive hub cluster

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 multicloud 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 Multicloud 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

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 [Restoring a hub cluster](#) from backup.
Important: Recovering a failed hub to its passive instance will only restore 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. Submariner is automatically installed once the managed clusters are imported on the passive hub.
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 will not be 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. 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.
 8. 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](#).
 9. 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	
[...]					

Disaster recovery with stretch cluster for Fusion Data Foundation

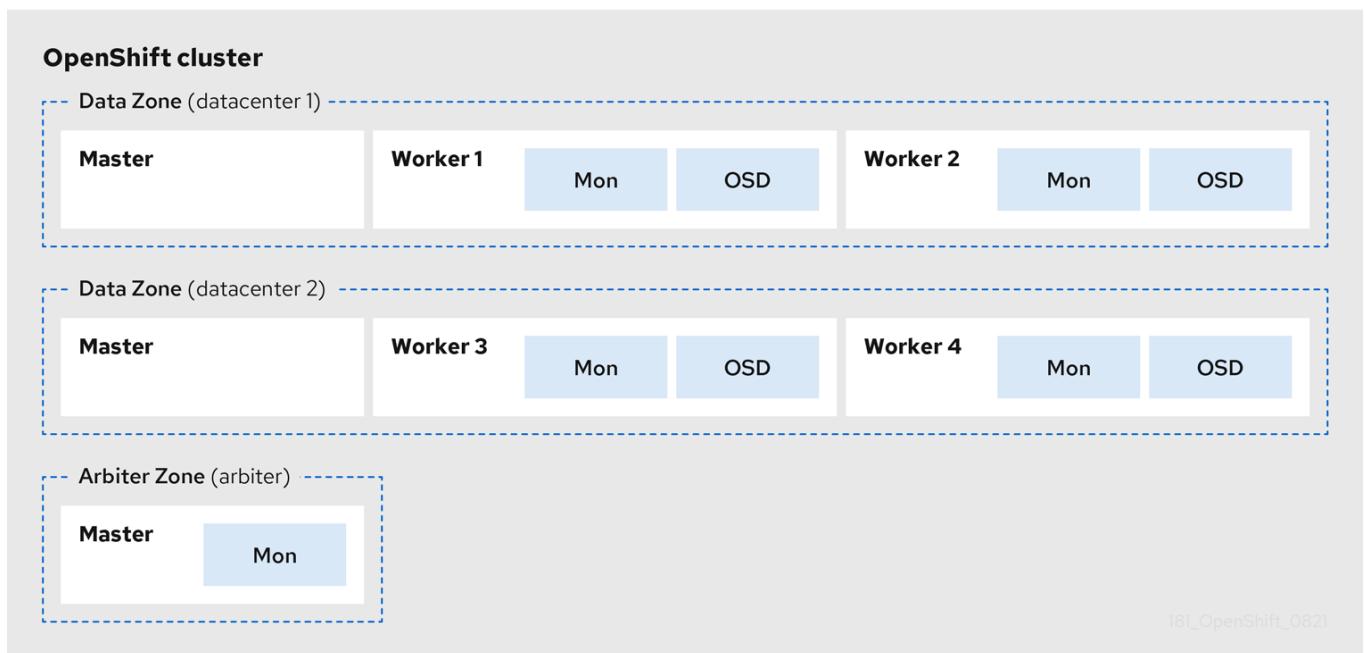
This section of the guide provides you with insights into the IBM Storage Fusion Data Foundation Disaster Recovery (DR) solution along with necessary configuration and recovery steps for stretch clusters.

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 have network connectivity with the OpenShift Container Platform nodes in the other two zones.

- [Requirements for enabling stretch cluster](#)
- [Applying topology zone labels to OpenShift Container Platform nodes](#)

- [Installing Local Storage Operator](#)
 - [Installing IBM Storage Fusion Data Foundation Operator](#)
 - [Creating Fusion Data Foundation cluster](#)
 - [Verifying Fusion Data Foundation deployment](#)
 - [Installing Zone Aware Sample Application](#)
 - [Recovering Fusion Data Foundation stretch cluster](#)
-

Requirements for enabling stretch cluster

- 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

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

About this task

Install the Local Storage Operator from the Operator Hub before creating IBM Storage 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 Storage Fusion Data Foundation Operator

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 Storage 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. Navigate to Installed Operators and verify that the Fusion Data Foundation Operator shows a green tick indicating successful installation.
 - c. Navigate 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

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 on 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.
 - 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, follow the steps below:
 - 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 above.
 - 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 **NetworkAttachmentDefinition**, 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 OpenShift Data Foundation deployment](#).

What to do next

[Installing Zone Aware Sample Application](#)

Verifying Fusion Data Foundation deployment

To ensure that Fusion Data Foundation is deployed correctly:

- [Verifying the state of the pods](#)
 - [Verifying the Fusion Data Foundation cluster is healthy](#)
 - [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

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
Fusion Data Foundation Operator	<ul style="list-style-type: none"> • 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)

Component	Corresponding pods
Multicloud Object Gateway	<ul style="list-style-type: none"> • <code>noobaa-operator-*</code> (1 pod on any worker node) • <code>noobaa-core-*</code> (1 pod on any storage node) • <code>noobaa-db-pg-*</code> (1 pod on any storage node) • <code>noobaa-endpoint-*</code> (1 pod on any storage node)
MON	<code>rook-ceph-mon-*</code> (5 pods are distributed across 3 zones, 2 per data-center zones and 1 in arbiter zone)
MGR	<code>rook-ceph-mgr-*</code> (2 pods on any storage node)
MDS	<code>rook-ceph-mds-ocs-storagecluster-cephfilesystem-*</code> (2 pods are distributed across 2 data-center zones)
RGW	<code>rook-ceph-rgw-ocs-storagecluster-cephobjectstore-*</code> (2 pods are distributed across 2 data-center zones)
CSI	<ul style="list-style-type: none"> • <code>cephfs</code> <ul style="list-style-type: none"> • <code>csi-cephfsplugin-*</code> (1 pod on each worker node) • <code>csi-cephfsplugin-provisioner-*</code> (2 pods distributed across worker nodes) • <code>rbd</code> <ul style="list-style-type: none"> • <code>csi-rbdplugin-*</code> (1 pod on each worker node) • <code>csi-rbdplugin-provisioner-*</code> (2 pods distributed across worker nodes)
Rook-ceph-crashcollector	<code>rook-ceph-crashcollector-*</code> (1 pod on each storage node and 1 pod in arbiter zone)
OSD	<ul style="list-style-type: none"> • <code>rook-ceph-osd-*</code> (1 pod for each device) • <code>rook-ceph-osd-prepare-ocs-deviceset-*</code> (1 pod for each device)

Verifying the Fusion Data Foundation cluster 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.
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

To know more 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 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

To know more 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-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 a 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 Red Hat OpenShift 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 IBM Storage 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
```

2. Deploy the example PHP application called `file-uploader`.

```
oc new-app openshift/php:7.3-ubi8~https://github.com/christianh814/openshift-php-upload-demo --name=file-uploader
```

Example Output:

```
Found image 4f2dcc0 (9 days old) in image stream "openshift/php" under tag "7.2-ubi8" for "openshift/php:7.2-ubi8"

Apache 2.4 with PHP 7.2
-----
PHP 7.2 available as container is a base platform for building and running various PHP 7.2 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, php72, php-72

* A source build using source code from https://github.com/christianh814/openshift-php-upload-demo will be created
* 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-registry.svc:5000/openshift/php@sha256:d97466f33999951739a76bce922ab17088885db610c0e05b593844b41d5494ea
STEP 1: FROM image-registry.openshift-image-registry.svc:5000/openshift/php@sha256:d97466f33999951739a76bce922ab17088885db610c0e05b593844b41d5494ea
STEP 2: LABEL "io.openshift.build.commit.author"="Christian Hernandez <christian.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
2f7f7b6b6f93396ffdf34cb2"
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](#)
- [Modifying deployment to be Zone Aware](#)

Scaling the application after installation

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 above 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
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 **ReadWriteOnce** (RWO) PV, the pods may get colocated on the same node.

Modifying deployment to be Zone Aware

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 [Controlling pod placement by using pod topology spread constraints](#).

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 here
          containers: # <-- 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

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](#)
- [Recovering zone-aware HA applications with rwx storage](#)
- [Recovering HA applications with rwx storage](#)
- [Recovering applications with RWO storage](#)
- [Recovering StatefulSet pods](#)

Understanding zone failure

In this section, zone failure is considered as a failure where all Red Hat 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.

Recovering zone-aware HA applications with rwx storage

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.

Recovering HA applications with rwx storage

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 Red Hat 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).

Recovering applications with RWO storage

Applications that use `ReadWriteOnce (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. Once 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-ae0-6e547f5f5a7c`).

```
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-m28bt-worker-d2hdm
  Warning FailedAttachVolume 4m5s  attachdetach-controller  Multi-Attach error for volume "pvc-0595a8d2-683f-443b-ae0-6e547f5f5a7c" Volume is already exclusively attached to one node and can't be attached to another
```

Recovering StatefulSet pods

Pods that are part of a StatefulSet have a similar issue as pods mounting `ReadWriteOnce (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, Red Hat 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](#)
- [Enabling disaster recovery dashboard on Hub cluster](#)
- [Viewing health status of disaster recovery replication relationships](#)
- [Disaster recovery metrics](#)
- [Disaster recovery alerts](#)

Enable monitoring for disaster recovery

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

Before you begin

- Ensure that you have already installed the following
 - OpenShift Container Platform version 4.15 and have administrator privileges.
 - ODF Multicluster Orchestrator with the console plugin enabled.
 - Red Hat Advanced Cluster Management for Kubernetes 2.10 (RHACM) from Operator Hub. For instructions on how to install, see [Installing RHACM](#).
- Ensure you have enabled observability on RHACM. See [Enabling observability guidelines](#).

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 [Adding custom metrics](#). 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 will start collecting the listed OpenShift 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

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. Navigate 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.

Disaster recovery metrics

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

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
- `scheduling_interval`: Scheduling interval value from DRPolicy

Metric value

The value is taken from `lastGroupSyncBytes` from DRPC status.

Disaster recovery alerts

This section provides a list of all supported alerts associated with IBM Storage Fusion Data Foundation within disaster recovery environment.

Recording rules

- Record: `ramen_sync_duration_seconds`

Expression

```
sum by (obj_name, obj_namespace, obj_type, job, policymame) (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(policymame, 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:
  cluster: '{{ $labels.cluster }}'
  severity: critical
annotations:
  description: >-
    Syncing of volumes (DRPC: {{ $labels.obj_name }}, Namespace: {{ $labels.obj_namespace }}) is taking more than thrice the scheduled
    snapshot interval. This may cause data loss and a backlog of replication
    requests.
  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:
  cluster: '{{ $labels.cluster }}'
  severity: critical
annotations:
  description: >-
    Syncing of volumes (DRPC: {{ $labels.obj_name }}, Namespace: {{ $labels.obj_namespace }}) is taking more than twice the scheduled snapshot interval. This may cause data loss and a backlog of replication requests.
  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](#)

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>
```

Once 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 will be 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

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](#)
- [volsync-rsync-src pod is in error state as it is unable to resolve the destination hostname](#)
- [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.

rbd-mirror daemon health is in warning state

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
--------	------------

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 <code>rbd-mirror</code> pod to a different node by simply deleting the pod and verify that it has been rescheduled on another node.
<code>leader: true</code> 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

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 `PlacementDecision` 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
existing PlacementDecision determined at step (2)
    cluster.open-cluster-management.io/decision-group-name: ""
    cluster.open-cluster-management.io/placement: cephfs-appset-busybox10-placement
  name: <placement-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 <placement-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 <placement-name>-decision-<n>
```

ApplicationSets that are DR protected, with a regional DRPolicy, stops firing the **VolumeSynchronizationDelay** alert.

BZ reference: [\[2268594\]](#)

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](#)

Recovering workload pods stuck in ContainerCreating state post zone recovery

Problem

After performing complete zone failure and recovery, the workload pods are sometimes stuck in **ContainerCreating** state with the any of the below 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 above mentioned 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

Troubleshooting

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 Storage Fusion customers might encounter.
- [How to provide feedback](#)
- [Events and error codes message references in IBM Storage Fusion](#)
This reference information provides a consolidated view of all the events or error messages that can be displayed when the IBM Storage Fusion system might results in an event or an error.
- [Troubleshooting issues in IBM Storage Fusion](#)
This topic covers the most common issues and their workarounds, which you might encounter while you work with the IBM Storage Fusion software.

Contacting IBM Support Center

The IBM® Support Center is available for various types of IBM software problems that IBM Storage Fusion customers might encounter.

Before you begin

Points to note when you contact the IBM Support Center:

1. For IBM Storage 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 Storage 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

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. Please share how we can improve your IBM Documentation experience to better meet your needs.

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. Please share how we can improve our product documentation to better meet your needs.

If you require support and a response to technical questions, please visit IBM's [technical support site](#) to open a defect.

Events and error codes message references in IBM Storage Fusion

This reference information provides a consolidated view of all the events or error messages that can be displayed when the IBM Storage 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 Storage 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 Storage 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 **EMY** 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 **EMYSTxxxx** in the message code.
- The list of events that can open tickets tag is associated to **EMYXC** in the message code.

An example of a message prefix format is **EMYST0001**.

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 Storage 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.

- [Data Cataloging events and error codes](#)

List of all the events that you might encounter while you work with Data Cataloging for the IBM Storage Fusion.

- [Backup & Restore events and error codes](#)

List of all the events that you might encounter during Backup & Restore for the IBM Storage Fusion.

- [Connection services events and error codes](#)

List of all the events that you might encounter during connection service for the IBM Storage 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.
 - **BMYPC5010**
IBM Storage Fusion operator installation is in progress.
 - **BMYPC6006**
A pod in the `openshift-operator-lifecycle-manager` namespace is not ready.
 - **BMYPC6007**
A `Catalogsource` is not ready.
 - **BMYPC6008**
Inaccessible registry.
 - **BMYPC6010**
IBM Storage Fusion operator is in a `Failed` state or not found.
 - **BMYPC6013**
The DNS resolution for image registry failed.
-

BMYIN1501

Application has been created successfully.

Severity

Informational

User response

No action is required.

BMYIN1502

Installing service Global Data Platform.

Severity

Informational

User action

No action is required.

BMYIN1503

Installed service Global Data Platform.

Severity

Informational

User action

No action is required.

BMYIN1504

Installing service Backup & Restore.

Severity

Informational

User action

No action is required.

BMYIN1505

Installed service Backup & Restore.

Severity

Informational

User action

No action is required.

BMYIN1506

Installing service Data Foundation.

Severity

Informational

User action

No action is required.

BMYIN1507

Installed service Data Foundation.

Severity

Informational

User action

No action is required.

BMYIN1508

Service is installed.

Severity

Informational

User response

No action is required.

BMYIN1509

Service installation is complete.

Severity

Informational

User response

No action is required.

BMYIN1510

Service instance is discovered.

Severity

Informational

User response

No action is required.

BMYIN1512

Catalog source is created.

Severity

Informational

User response

No action is required.

BMYIN1513

Catalog source creation is completed.

Severity

Informational

User response

No action is required.

BMYIN1514

Discovered service Data Foundation.

Severity

Informational

User action

No action is required.

BMYIN2501

Error installing service Global Data Platform.

Severity

Warning

User action

Collect [Storage logs](#) and contact [IBM support](#).

BMYIN2502

Error installing service Backup & Restore.

Severity

Warning

User action

Collect [Backup & Restore logs](#) and contact [IBM support](#).

BMYIN2503

Error installing service Data Foundation.

Severity

Warning

User action

Collect [Data Foundation logs](#) and contact [IBM support](#).

BMYIN3116

Service is Unhealthy.

Severity

Critical

User action

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 Storage Fusion.
 - Go to Services page in IBM Storage 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 action

Do the following steps to diagnose and report the problem:

1. Run the following command and try to debug the results.

```
oc describe catalogsource <catalogsource-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 action

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 Storage Fusion.
 - Go to Services page in IBM Storage Fusion user interface.
 - Click the ellipses menu for the corresponding service.
 - Select the Collect logs button. It downloads all the necessary logs.

BMYPC5010

IBM Storage Fusion operator installation is in progress.

Severity

Error

Problem description

The IBM Storage 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 Storage Fusion operator or other IBM Storage Fusion components ability to install, upgrade, or function.

Recommended actions

Do the following steps to resolve the issue:

1. Check for IBM Storage 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 Storage Fusion operator are as follows:
 - Check whether IBM Storage Fusion operator is installing or upgrading.
 - Some changes might corrected in any deployment spec in CSV for IBM Storage Fusion operator. It recreates pods and reconciles to succeeded state in some time.
 - One of the IBM Storage 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 Storage Fusion operator is successful.

BMYPC6006

A pod in the `openshift-operator-lifecycle-manager` namespace is not ready.

Severity

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 Storage Fusion HCI System.

Recommended 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 Storage 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 Storage 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

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.

Recommended 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. Make sure 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 Storage 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 Storage 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

Error

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 Storage 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.

Recommended 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.

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 Storage 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 Storage Fusion is installed.

```
oc create configmap precheck-acks -n $FUSION_NS
```

3. In the `precheck-acks` config map, add the key `RegistryAccessibility` and set the value to `true`.

```
oc patch configmap precheck-acks -n $FUSION_NS --type merge -p '{"data": {"RegistryAccessibility": "true"}}'
```

BMYPC6010

IBM Storage Fusion operator is in a **Failed** state or not found.

Severity

Error

Problem description

It indicates that the IBM Storage Fusion operator is not in a **Failed** state or not found. It can block the ability of the IBM Storage Fusion operator or other IBM Storage Fusion components ability to install, upgrade, or do other function.

Recommended actions

Do the following steps to resolve the issue:

1. Check for IBM Storage 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 Storage Fusion operator are as follows:
 - One of the IBM Storage 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 Storage 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 Storage 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 Storage 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 Storage Fusion components and the OpenShift® Container Platform. Failure to address the registry access can result in an **ImagePullbackOff** error of the pod.

Recommended 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 Storage 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 Storage 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"}}'
```

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 Storage 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.

BMYUP1101

Upgrade of Software components is in progress.

Severity

Informational

User action

No action is required.

BMYUP1102

Software upgrade is complete.

Severity

Informational

User action

No action is required.

BMYUP1103

Upgrade of Red Hat® OpenShift® cluster is in progress.

Severity

Informational

User action

No action is required.

BMYUP1104

Red Hat® OpenShift® cluster upgrade is complete.

Severity

Informational

User action

No action is required.

BMYUP1111

Started upgrade of the Global Data Platform service.

Severity

Informational

User response

No action is required.

BMYUP1112

Completed upgrade of the Global Data Platform service.

Severity

Informational

User response

No action is required.

BMYUP1113

Started upgrade of the Data Protection service.

Severity

Informational

User response

No action is required.

BMYUP1114

Completed upgrade of the Data Protection service.

Severity

Informational

User response

No action is required.

BMYUP1115

Started upgrade of the Data Foundation service.

Severity

Informational

User response

No action is required.

BMYUP1116

Completed upgrade of the Data Foundation service.

Severity

Informational

User response

No action is required.

BMYUP1117

IBM Storage Fusion service upgrade available for the on-boarded service.

Severity

Informational

User response

No action is required.

BMYUP1118

Service upgrade is available.

Severity

Informational

User response

No action is required.

BMYUP1119

Service upgrade in progress.

Severity

Informational

User response

No action is required.

BMYUP1120

Service upgrade is completed.

Severity

Informational

User response

No action is required.

BMYUP1121

Operator Upgrade in progress.

Severity

Informational

User response

No action is required.

BMYUP1123

Operator Upgrade completed.

Severity

Informational

User response

No action is required.

BMYUP3101

Failed to upgrade the Global Data Platform service.

Severity

Critical

User response

Collect [Global Data Platform logs](#) and contact [IBM support](#).

BMYUP3102

Failed to upgrade Backup & Restore service.

Severity

Critical

User action

Do the following steps to diagnose and report the problem:

1. Ensure that you performed the before you begin steps of [Upgrading IBM Storage Fusion](#).
2. If the issue persists, contact [IBM support](#).

BMYUP3103

Failed to upgrade the Data Foundation service.

Severity

Critical

User response

Collect [Data Foundation logs](#) and contact [IBM support](#).

BMYUP3104

Service upgrade is failing.

Severity

Critical

User action

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 Storage Fusion.
 - Go to Services page in IBM Storage 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 response

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](#).

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 is 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 response

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 response

Collect [Global Data Platform logs](#) and contact [IBM support](#).

BMYSS0003

IBM Storage Scale storage installation succeeded.

Severity

Informational

User response

No action is required.

BMYSS0004

IBM Storage Scale storage upgrade is in progress.

Severity

Informational

User response

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 response

Collect [Global Data Platform logs](#) and contact [IBM support](#).

BMYSS0006

IBM Storage Scale storage upgrade succeeded.

Severity

Informational

User response

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 response

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 response

Collect [Global Data Platform logs](#) and contact [IBM support](#).

BMYSS0010

IBM Storage Scale operator status is 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 response

Collect [Global Data Platform logs](#) and contact [IBM support](#).

BMYSS0011

IBM Storage Scale remote mount file system installation is in progress.

Severity

Informational

User response

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

W

User response

Collect [Global Data Platform logs](#) and contact [IBM support](#).

BMYSS0013

IBM Storage Scale remote mount file system installation succeeded.

Severity

Informational

User response

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.

BMYSS0100

Data Foundation installation is in progress.

Severity

Informational

User response

No action is required.

BMYSS0101

Data Foundation installation succeeded.

Severity

Informational

User response

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 response

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 response

Collect [Data Foundation logs](#) and contact [IBM support](#).

BMYSS0104

Data Foundation status is healthy.

Severity

Informational

User response

No action is required.

BMYSS0105

Data Foundation upgrade is in progress.

Severity

Informational

User response

No action is required.

BMYSS0106

Data Foundation upgrade succeeded.

Severity

Informational

User response

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 response

Collect [Data Foundation logs](#) and contact [IBM support](#).

BMYSS0108

Data Foundation storage cluster installation is in progress.

Severity

Informational

User response

No action is required.

BMYSS0109

Data Foundation storage cluster installation succeeded.

Severity

Informational

User response

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 response

Collect [Data Foundation logs](#) and contact [IBM support](#).

BMYSS0111

Data Foundation storage cluster status is healthy.

Severity

Informational

User response

No action is required.

BMYSS0112

Data Foundation is scaling up automatically.

Severity

Informational

User response

No action is required.

BMYSS0115

Field <fieldname> is required.

Severity

Error

User response

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 response

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 response

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 response

Ensure that the user must be provided with both a client certificate and a client key, or not provide either of them.

EMYSS0119

Field <fieldname> is immutable.

Severity

Error

User response

User cannot update the field <fieldname>.

EMYSS0120

KMS server is not reachable.

Severity

Error

User response

Ensure that you need to check connectivity of KMS server.

EMYSS0121

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 response

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](#).

EMYSS0122

Data Foundation capacity is adding.

Severity

Informational

User response

No action is required.

EMYSS0125

Data Foundation capacity is added.

Severity

Informational

User response

No action is required.

BMYSS0126

Nodes are added to Data Foundation storage cluster successfully.

Severity

Informational

User response

No action is required.

BMYSS0127

Data Foundation is auto-scaled up successfully.

Severity

Informational

User response

No action is required.

Data Cataloging events and error codes

List of all the events that you might encounter while you work with Data Cataloging for the IBM Storage Fusion.

For Informational, Warning, and Critical events, see the following:

- [**BMYDC0001**](#)
Data Cataloging service is healthy.
- [**BMYDC0002**](#)
Data Cataloging service health is degraded.
- [**BMYDC0003**](#)
Data Cataloging service workload resources are ready.
- [**BMYDC0004**](#)
Data Cataloging service workload resources are not ready.
- [**BMYDC0005**](#)
Data Cataloging service API is available.
- [**BMYDC0006**](#)
Data Cataloging service API is not completely available.
- [**BMYDC0100**](#)
Data Cataloging service installation in progress.
- [**BMYDC0101**](#)
Data Cataloging service installation succeeded.
- [**BMYDC0102**](#)
Data Cataloging service installation failing.
- [**BMYDC0103**](#)
Data Cataloging service is already installed.
- [**BMYDC0104**](#)
Data Cataloging service upgrade in progress.
- [**BMYDC0105**](#)
Data Cataloging service upgrade succeeded.
- [**BMYDC0106**](#)
Data Cataloging service upgrade failing.
- [**BMYDC0107**](#)
Data Cataloging service is not ready for upgrade.
- [**BMYDC0108**](#)
New version is available for Data Cataloging service upgrade.

- [**BMYDC0109**](#)
No new versions available for Data Cataloging service upgrade.
- [**BMYDC0110**](#)
Data Cataloging service prerequisites are ready.
- [**BMYDC0113**](#)
Data Cataloging service Kafka instances are ready.
- [**BMYDC0114**](#)
Data Cataloging service Kafka instances are not ready.
- [**BMYDC0115**](#)
Data Cataloging service Db2 is ready.
- [**BMYDC0116**](#)
Data Cataloging service Db2 is not ready.
- [**BMYDC0117**](#)
Data Cataloging service license was not accepted.
- [**BMYDC0118**](#)
Cluster API that is required by the Data Cataloging service is not available.

BMYDC0001

Data Cataloging service is healthy.

Severity

Informational

User response

No action is required.

BMYDC0002

Data Cataloging service health is degraded.

Severity

Warning

User response

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

Data Cataloging service workload resources are ready.

Severity

Informational

User response

No action is required.

EMYDC0004

Data Cataloging service workload resources are not ready.

Severity

Warning

User response

Do the following steps to diagnose and report the problem:

- 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-scalefmdata mover	1/1	1	1	6d11h
isd-scaleilmdata mover	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-wkconnector	1/1	1	1	6d11h

- If you find any issue with the specific components listed, collect the [Data Cataloging logs](#) and contact [IBM support](#).

EMYDC0005

Data Cataloging service API is available.

Severity

Informational

User response

No action is required.

EMYDC0006

Data Cataloging service API is not completely available.

Severity

Warning

User response

Do the following steps to diagnose and report the problem:

- 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
```

- If you find any issue with the specific components listed, collect the [Data Cataloging logs](#) and contact [IBM support](#).

EMYDC0100

Data Cataloging service installation in progress.

Severity

Informational

User response

No action is required.

EMYDC0101

Data Cataloging service installation succeeded.

Severity

Informational

User response

No action is required.

EMYDC0102

Data Cataloging service installation failing.

Severity

Critical

User response

Do the following steps to diagnose and report the problem:

- 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
```

- 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.

EMYDC0103

Data Cataloging service is already installed.

Severity

Informational

User response

No action is required.

BMYDC0104

Data Cataloging service upgrade in progress.

Severity

Informational

User response

No action is required.

BMYDC0105

Data Cataloging service upgrade succeeded.

Severity

Informational

User response

No action is required.

BMYDC0106

Data Cataloging service upgrade failing.

Severity

Critical

User response

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

Data Cataloging service is not ready for upgrade.

Severity

Informational

User response

No action is required.

BMYDC0108

New version is available for Data Cataloging service upgrade.

Severity

Informational

User response

No action is required.

BMYDC0109

No new versions available for Data Cataloging service upgrade.

Severity

Informational

User response

No action is required.

BMYDC0110

Data Cataloging service prerequisites are ready.

Severity

Informational

User response

No action is required.

BMYDC0113

Data Cataloging service Kafka instances are ready.

Severity

Informational

User response

No action is required.

BMYDC0114

Data Cataloging service Kafka instances are not ready.

Severity

Warning

User response

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

Data Cataloging service Db2 is ready.

Severity

Informational

User response

No action is required.

BMYDC0116

Data Cataloging service Db2 is not ready.

Severity

Warning

User response

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

Data Cataloging service license was not accepted.

Severity

Critical

User response

Do the following steps to diagnose and report the problem:

1. This is not expected, by default Data Cataloging license is accepted.
2. Run the following command to check the 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 Data Cataloging service is not available.

Severity

Critical

User response

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 (4.10, 4.12, 4.13 and 4.14).

Backup & Restore events and error codes

List of all the events that you might encounter during Backup & Restore for the IBM Storage 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 response

Collect [Backup and Restore logs](#) and contact [IBM support](#).

BMYBR0002

The job was not started because of an already running job.

Severity

Informational

User response

No action is required.

BMYBR0003

There was an error when processing the job in the service.

Severity

Critical

User response

Collect [Backup and Restore logs](#) and contact [IBM support](#).

BMYBR0009

There was an error when processing the job in the service.

Severity

Critical

User response

Collect [Backup and Restore logs](#) and contact [IBM support](#).

BMYDP1146

Policy assignment is created initial backup for an application.

Severity

Informational

User response

No action is required.

BMYDP1163

Completed backup for application from backup policy.

Severity

Informational

User response

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 Storage 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 response

No action is required.

BMYCS101

Create bootstrap secret completed.

Severity

Informational

User response

No action is required.

BMYCS102

Create a kubeconfig secret in progress.

Severity

Informational

User response

No action is required.

BMYCS103

Create kubeconfig secret completed.

Severity

Informational

User response

No action is required.

BMYCS300

Init secret has the same API endpoint as the local cluster.

Severity

Critical

User response

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 response

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 response

Collect [Storage logs](#) and contact [IBM support](#).

BMYCS303

Failed to get CA certificate from the remote cluster.

Severity

Critical

User response

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 response

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 response

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 response

Collect [Storage logs](#) and contact [IBM support](#).

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 response

Do the following steps to resolve the error:

1. Delete the unnecessary logs through the IBM Storage Fusion user interface to get storage space. For more information, see [Delete a log package](#).
 2. If the issue persists, then contact the [IBM support](#).
-

BMYLC0002

Remaining storage space is less than 10 percent.

Severity

User response

Do the following steps to resolve the error:

1. Delete the unnecessary logs through the IBM Storage Fusion user interface to get storage space. For more information, see [Delete a log package](#).
 2. If the issue persists, then contact the [IBM support](#).
-

Troubleshooting issues in IBM Storage Fusion

This topic covers the most common issues and their workarounds, which you might encounter while you work with the IBM Storage Fusion software.

- [Installation and upgrade issues](#)
Use the troubleshooting tips and tricks in IBM Storage Fusion installation and upgrade.
 - [IBM Storage Fusion user interface issues](#)
A list of all known issues in the IBM Storage Fusion user interface.
 - [Troubleshooting installation and upgrade issues in IBM Storage Fusion services](#)
Use these troubleshooting information to know install and upgrade problems related to IBM Storage Fusion services.
 - [Troubleshooting Backup & Restore service issues](#)
List of known Backup & Restore issues in IBM Storage Fusion.
 - [Troubleshooting Global Data Platform issues](#)
 - [Troubleshooting IBM Fusion Data Foundation service](#)
Use these troubleshooting information to know the problem and workaround when install or configure IBM Fusion Data Foundation service.
 - [IBM Storage Fusion Data Cataloging known issues](#)
List of all troubleshooting and known issues exist in version 2.1 of Data Cataloging.
-

Installation and upgrade issues

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

Update operator OOMKilled error

Resolution

To resolve the OOMKilled issue for the update operator, do the following resolution steps:

Note: If the pod is in a CrashLoopBackOff state, delete the `isf-update-operator-*`

pod. It comes back to running state after a couple of minutes. Do steps 1-4 in the following resolution steps when the update operator pod is in Running state.

1. In the OpenShift® Container Platform console, go to Home > Search.
2. Search for `UpdateManager` in the Resources drop-down list.
3. In the `UpdateManagers`, open the `version` instance.
4. Go to the YAML tab.
5. Increase the memory limit in `spec.resources.limits.memory`.
6. After a couple of minutes, check whether the IBM Storage Fusion `clust erserviceversion` object (Operators > Installed Operators > IBM Storage Fusion operator > YAML tab) reflects the updated limit set for the update operator:
 - Search for the deployment name of the update operator (`isf-update-operator-controller-manager`) from the list of deployments under `spec.install.spec.deployments`.
 - In the specified deployment object, search for the container name `manager` under the `spec.template.spec.containers`. Also, check whether the `limits.memory` is equal to the one in the `UpdateManager` CR. If not equal, change the `memory` under `limits` to the same limits value as mentioned in the update operator CR in step 5.
 - Go to Workloads > Deployments > `isf-update-operator-controller-manager` > YAML tab and check whether the `limits.memory` is equal to the limit set in the previous step. If not equal, change the `memory` under `limits` to the same limits value as mentioned in the previous steps.

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.

Installation stuck in the custom namespace

Problem statement

when you deployed IBM Storage Fusion on the `custom-ns`, the operator is stuck in installing state because the serviceability pod is not coming up.

Diagnosis

The operator fails to succeed as it continuously waits for the `isf-serviceability-operator` pod to come up. The serviceability pod requires cluster-admin privileges to start the pod as a non-root user. In the case of `custom-ns` installations, the prereq operator is expected to create this cluster-role binding. However, this code is not triggered until `SpectrumFusion` CR is created. As a result, the operator remains in the installed or failed state until policies are accepted from the IBM Storage Fusion user interface or the `SpectrumFusion` CR is created.

Resolution

As a resolution, move the code to create this rolebindings from the controller to `main.go` in the `isf-prereq-operator`. It creates the CRB as soon as prereq operator pod comes up.

IBM Storage Fusion user interface issues

A list of all known issues in the IBM Storage 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.

Troubleshooting installation and upgrade issues in IBM Storage Fusion services

Use these troubleshooting information to know install and upgrade problems related to IBM Storage Fusion services.

- [Global Data Platform service issues](#)
Use this troubleshooting information to resolve install and upgrade problems that are related to Global Data Platform service.
- [Backup & Restore service install and upgrade issues](#)
Use this troubleshooting information to resolve install and upgrade problems that are related to Backup & Restore service.
- [Data Cataloging service issues](#)
Use this troubleshooting information to resolve install and upgrade problems that are related to the Data Cataloging service.
- [Common service installation issues](#)
Use this troubleshooting information to resolve common install and upgrade problems related to IBM Storage Fusion services.

Global Data Platform service 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.

IBM Storage Scale stuck with error

Problem statement

If IBM Storage Scale installation stuck with the error ERROR Failed to define vdiskset {"controller": "filesystem", "controllerGroup": "scale.spectrum.ibm.com", "controllerKind": "Filesystem", "Filesystem": {"name": "ibmspectrum-fs", "namespace": "ibm-spectrum-scale"}, "namespace": "ibm-spectrum-scale", "name": "ibmspectrum-fs", "reconcileID": "5fcf21a4-b488-44db-8db5-0f3c1ab695cd", "cmd": "/usr/lpp/mmf/bin/mmvdisk vs define --vs ibmspectrum-fs-0 --rg rg1 --code 4+2P --bs 4M -

Resolution

As a resolution, run the following command to manually create a recovery group.

```
mmvdisk recoverygroup create --recovery-group <RG name> --node-class <Node class name>
```

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 stuck, 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 ReadWriteOnce is 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: mmfsevn: Module mmfslinux is still in use.
compute-13.ibm-spectrum-scale-core.ibm-spectrum-scale.svc.cluster.local: mmfsevn: 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`.

IBM Storage Scale upgrade gets stuck

Cause

The IBM Storage Scale upgrade gets stuck because of `isf-storage-service`. It happens whenever the `scaleoutstatus` or `scaleupStatus` is **In Progress** in the Scale CR.

Important: Do not update the latest image for the `isf-storage-service` when a IBM Storage Scale operation is in progress.

Workaround

Steps to upgrade the IBM Storage Scale on the rack:

1. Go to the `isf-storage-service` in `ibm-spectrum-fusion-ns` project and change the image to `cp.icr.io/cp/isf/isf-storage-services:2.2.0-latest` or run the following OC command.

```
oc patch deploy isf-storage-service-dep -n ibm-spectrum-fusion-ns --patch='{"spec": {"template": {"spec": {"containers": [{"name": "app", "image": "cp.icr.io/cp/isf/isf-storage-services:2.2.0-latest"}]}}}}'
```

2. After the `isf-storage-service` pod goes to running state and points to the `cp.icr.io/cp/isf/isf-storage-services:2.2.0-latest` version, run the following API endpoint command in the `isf-storage-service` pod's terminal:

```
curl -k https://isf-scale-svc/api/v1/upgradeExcludingOperator
```

You can observe the new logs for `isf-storage-service` as well.

3. Set the Edition to `erasure-code` in the Daemon

CR:

```
oc patch daemon ibm-spectrum-scale \
--type='json' -n ibm-spectrum-scale \
-p="[{ 'op': 'replace', 'path': '/spec/edition', 'value': 'erasure-code'}]"
```

4. Run the following curl command to deploy the operator on the Red Hat® OpenShift® Container Platform cluster:

```
curl -k https://isf-scale-svc/api/v1/upgradeWithOperator
```

You can observe new logs for `isf-storage-service`.

5. Wait for sometime and check whether the status of IBM Storage Scale core pod is in restart or running state in `ibm-spectrum-scale`.

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.

Scale installation might get stuck with ECE pods in `init` state

Problem statement

The Scale installation might get stuck with ECE pods in `init` state with the following error:

The node already belongs to a GPFS cluster.

Resolution

1. Check whether the daemon-network IP of the pod is pingable.

2. If it is not pingable, then restart all available TORs.
 3. After you restart TORs, check whether daemon-network IP is pingable.
 4. Run the following command to manually clean up the nodes:
For the given worker node, run the following **oc** command:
- ```
oc debug node/<openshift_worker_node> -T -- chroot /host sh -c "rm -rf /var/mmfss"
```
5. Kill the pod and wait for it to come up again. The Scale installation resumes after all the ECE pods goes to running state.

## Expansion rack fails for 4+2p setup

---

### Problem statement

Sometimes, in a high-availability cluster, the expansion rack fails for 4+2p setup.

### Resolution

- Do the following workaround steps to resolve the issue:
  1. In the OpenShift Container Platform user interface, go to Administration > CustomResourceDefinitions > Scale > Instances > storagemanager > Yaml and check whether the **scaleoutStatus: IN-PROGRESS**
  2. If **scaleoutStatus: IN-PROGRESS**, go to Administration > CustomResourceDefinitions > RecoveryGroup > Instances and check for the created RecoveryGroups.
  3. If more than two recovery groups are created, do the following actions:
    - a. Go to Administration > CustomResourceDefinitions > RecoveryGroup > Instances > ...
    - b. Find **RecoveryGroup** instance with suffix rg2.
    - c. Click the ellipsis menu for that instance and select the Delete RecoveryGroup option.
  4. Do the following checks to validate whether the node upsize completed successfully:
    - In the IBM Storage Fusion user interface, go to the IBM Storage Scale user interface by using the app switcher.
    - Click Recovery groups on the IBM Spectrum Scale RAID tile.
    - Click the entry of Recovery group server nodes.
    - Check whether all the newly added nodes for expansion rack are listed and is in Healthy state.
    - Create a sample PVC to check whether the storage can be provisioned.
  5. Check for the labels on each recovery group.
    - a. In the OpenShift Container Platform console, go to Administration > CustomResourceDefinitions > RecoveryGroup > Instances > Yaml.
    - b. Check whether the YAML contains either of these two labels:
      - **scale.spectrum.ibm.com/scale: up**
      - **scale.spectrum.ibm.com/scale: out**
    - c. If it is yes for the previous step, remove those labels from each recovery group and click Save.

## Known issues in installation

---

- The display unit of filesystem block size in **storagesetupcr** is MiB and that of **ibmspectrum-fs** is M. As IBM Storage Scale also uses MiB format internally, you can ignore this inconsistency.

## Known issues in upgrade

---

- During upgrade, the IBM Storage 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 Storage Fusion user interface in upgrade details page.
- During the upgrade, an intermittent error shows on the IBM Storage Fusion user interface that the progress percentage for the Global Data Platform decreased. It occurs, especially when the upgrade for one node is completed.

## Backup & Restore service install and upgrade issues

---

Use this troubleshooting information to resolve install and upgrade problems that are related to Backup & Restore service.

## Application information not repopulated after service re-installation

---

### Problem statement

The application information gets removed during Backup & Restore service uninstallation. It does not get repopulated in MongoDB after the service re-installation. 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 Storage 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 Storage 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 navigate 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 Storage 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 Crashloopbackoff state.

### Resolution

In the resource settings of `guardian-dp-operator` pod that resides in `ibm-backup-restore` namespace, set the value of IBM Storage Fusion operator memory limits to 1000mi.

Example:

```
resources:
 limits:
 cpu: 1000m
 memory: 1000Mi
 requests:
 cpu: 500m
 memory: 250Mi
```

## storage-operator pod crashes with CrashLoopBackOff status

---

### Problems statement

The storage-operator pod crashes due to OOM error.

### Resolution

To resolve the error, increase the memory limit from 300Mi to 500Mi.

1. Log in to the Red Hat® OpenShift® web console as an administrator.
2. Select the project `ibm-spectrum-fusion-ns`.
3. Select the `isf-storage-operator-controller-manager-xxxx` pod, and go to the YAML tab.
4. In the YAML, change the memory limit for `isf-storage-operator-controller-manager-xxxx` container from 300Mi to 500Mi.

```
containers:
 - resources:
 limits:
 cpu: 100m
 memory: 500Mi
```

5. Click Save.

## Backup & Restore service goes into unknown state

### Problem statement

This Backup & Restore service might goes into unknown state when you upgrade IBM Storage Fusion from 2.6.x to 2.8.0.

### Resolution

It automatically shows healthy on the IBM Storage Fusion user interface after you upgrade the Backup & Restore service.

## MongoDB pod crashes with CrashLoopBackOff status

### Problem statement

The MongoDB pod crashes due to 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 stuck at 5% during upgrade

### Diagnosis

1. In OpenShift user interface, go to the Installed Operator and filter on `ibm-backup-restore` namespace.
2. Click `IBM Storage Fusion Backup and Restore Server`.
3. Go to Subscriptions.
4. If you see the following errors, then do the workaround steps to resolve the issue:

```
"error validating existing CRs against new CRD's schema for "guardiancopyrestores.guardian.isf.ibm.com": error validating
```

### Resolution

1. From command line or command prompt, log in to the cluster and run the following commands to delete the `guardiancopyrestore` CRs and 2.6.0 cs:

```
oc -n ibm-backup-restore delete guardiancopyrestore.guardian.isf.ibm.com --all
oc -n ibm-backup-restore delete csv guardian-dm-operator.v2.6.0
```

2. From OpenShift Container Platform console, go to Installed Operator and filter on `ibm-backup-restore` namespace.
3. Click `IBM Storage Fusion Backup and Restore Server`.
4. Go to Subscriptions.
5. Find the failing `installplan` and delete it.
6. Go to Installed Operator and go to `ibm-backup-restore` namespace.
7. Find `IBM Storage Fusion Backup and Restore Server` and click Upgrade available and approve the Install plan for `IBM Storage Fusion Backup and Restore Server`.
8. Wait for the service upgrade to resume.

## Backup & Restore service installation gets stuck after upgrade to IBM Storage Fusion 2.7.0

### Problem statement

If the installation or upgrade of the Backup & Restore service does not reach 100% completion, then check for failed startup probes. Run the following command to look for any pods that are not in the READY state:

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

Example output:

| NAME                            | READY | STATUS  | RESTARTS      | AGE |
|---------------------------------|-------|---------|---------------|-----|
| applicationsvc-855746ffbf-2pmz7 | 0/1   | Running | 5 (2m8s ago)  | 17m |
| ...                             |       |         |               |     |
| job-manager-845dc56b8d-r5w6j    | 0/1   | Running | 5 (2m50s ago) | 17m |
| ...                             |       |         |               |     |

1. Go to OpenShift Container Platform.
2. Go to the Installed Operators view with the selected `ibm-backup-restore` project or namespace.
3. Click IBM Storage Fusion Backup & Restore Server.
4. Select the Data Protection Server tab and click the instance.
5. Select the YAML tab and update the following:

```
triggerUpgrade: false
```

```
to
triggerUpgrade: true
```

#### 6. Save and reload the YAML.

After some time, all of the pods must be READY and the Backup & Restore service must show healthy in IBM Storage Fusion user interface.

#### Diagnosis

If found, then describe each of the pods and look at the event section to determine if there is a startup probe failure.

Example output showing probe failure:

```
Events:
Type Reason Age From Message
---- ---- -- --- -----
...
Warning Unhealthy 3m40s (x5 over 6m40s) kubelet Startup probe failed: HTTP probe failed with statuscode:
```

If one or more pods show startup probe failures, then patch the probes to provide additional start time. Update the following script and run to patch the probes. To determine the respective deployment names, update the DEPLOYMENT\_NAMES variable with the list of deployments that have failing startup probes and run the script.

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

Example output:

| NAME           | READY | UP-TO-DATE | AVAILABLE | AGE |
|----------------|-------|------------|-----------|-----|
| ...            |       |            |           |     |
| applicationsvc | 0/1   | 1          | 1         | 12h |
| ...            |       |            |           |     |
| job-manager    | 0/1   | 1          | 1         | 12h |
| ...            |       |            |           |     |

Now, check whether the pods are online. If the startup probes continue to fail, then increase the STARTUP\_DELAY\_SEC parameter and try again.

If it is an upgrade and the pods come online, but the progress does not reach 100% in the IBM Storage Fusion user interface even after 30 minutes, then you must update the Backup & Restore Server CR to re-trigger the upgrade.

#### Resolution

Update the Backup & Restore Server CR to re-trigger the upgrade

## Backup & Restore service install 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 Storage 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.

## Data Cataloging service issues

---

Use this troubleshooting information to resolve install and upgrade problems that are related to the Data Cataloging service.

## 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
```

## 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 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
```

## 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 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
```

## Data Cataloging service is not installed successfully on IBM Storage Fusion HCI System 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 (UnsupportedMediaTypes): the body of the request was in an unknown format - accepted media types include: application/json-patch+json, application/merge-patch+json, 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 Storage Fusion operator.
- c. Select the IBM Storage 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.

## 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 Data Cataloging service. For procedure, see [Uninstalling Data Cataloging](#).
2. Create a file with the 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":{"tempsts":"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 Data Cataloging service from the IBM Storage Fusion user interface. For procedure, see [Data Cataloging](#).

---

## Common service installation issues

Use this troubleshooting information to resolve common install and upgrade problems related to IBM Storage Fusion services.

### ImagePull failure during installation or upgrade of any service

---

If an `ImagePull` failure occurs during the installation or upgrade of any service, then restart the pod and retry. If the issue persists, contact [IBM support](#).

### Rook-cephfs pods are in CrashLoopBackOff off state

---

#### Problem statement

Data Cataloging and Backup & Restore services that are installed, go into degraded state, and `rook-ceph-mds-ocs-storagecluster` pods are in `CrashLoopBackOff` state.

#### Resolution

Follow the steps to resolve the issue:

1. If IBM Storage Fusion is installed with OpenShift® Container Platform or Data Foundation v4.10.x, then you can upgrade it to v4.11.x.

2. The upgrade of OpenShift Container Platform or Data Foundation from v4.10.x to v4.11.x resolves the `CrashLoopBackOff` error in `rook-ceph-mds-ocs-storagecluster` pods and get it to running state. Eventually services also go to healthy state.  
Note: Also, you can upgrade OpenShift Container Platform or Data Foundation to further versions supported by IBM Storage Fusion.

## Troubleshooting common installation issues

---

- If you find an error saying `Configmap fusionplatform not found in fusion namespace` in the preparer operator logs, then the error does not have any impact and can be ignored from the `isf-prereq-operator-controller-manager-xxxx` pod logs
  - Problem statement  
Whenever the upgrade button is unavailable for any service
- Resolution  
During the offline upgrade, if you do not see the upgrade button for any of the services after upgrading IBM Storage Fusion operator, then check the `catalogsource` pod, and it should be in a running state. For any `imagepullbackoff` error, ensure you have completed mirroring and updated `imagecontentsourcepolicy`.

---

## Troubleshooting Backup & Restore service issues

List of known Backup & Restore issues in IBM Storage Fusion.

- [Backup issues](#)**  
List of backup issues in the Backup & Restore service of IBM Storage Fusion.
- [Restore issues](#)**  
List of restore issues in Backup & Restore service of IBM Storage Fusion.
- [Backup & Restore service install and upgrade issues](#)**  
Use this troubleshooting information to resolve install and upgrade problems that are related to Backup & Restore service.
- [IBM Cloud Pak for Data backup and restore issues](#)**  
This section lists the troubleshooting tips and tricks during backup and restore of IBM Cloud Pak for Data.
- [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](#)**  
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 Storage Fusion.

### 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 <https://www.ibm.com/docs/en/scalecsi/2.10?topic=snapshot-create-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 & Restore service deployed in IBM Cloud Satellite

---

#### Problem statement

You can encounter an error when you attempt backup operation on IBM Storage Fusion Backup & Restore service that is deployed in IBM Cloud® Satellite.

#### Diagnosis

Backup operations fail with the following log entries:

```
level=error msg="Error backing up item" backup=<item> error="error executing custom action (groupResource=pods, namespace=<namespace>, name=<name>): rpc error: code = Unknown desc = configmaps \"config\" not found" error.file="/remote-source/velero/app/pkg/backup/item_backupper.go:326" error.function="github.com/vmware-tanzu/velero/pkg/backup.(*itemBackupper).executeActions" logSource="/remote-source/velero/app/pkg/backup/backup.go:417" name=<name>
level=error msg="Error backing up item" backup=<item> error="error executing custom action (groupResource=replicaset.apps, namespace=<namespace>, name=<name>): rpc error: code = Unknown desc = configmaps \"config\" not found" error.file="/remote-source/velero/app/pkg/backup/item_backupper.go:326" error.function="github.com/vmware-tanzu/velero/pkg/backup.(*itemBackupper).executeActions" logSource="/remote-source/velero/app/pkg/backup/backup.go:417" name=<name>
level=error msg="Error backing up item" backup=<item> error="error executing custom action (groupResource=deployments.apps, namespace=<namespace>, name=<name>): rpc error: code = Unknown desc = configmaps \"config\" not found" error.file="/remote-source/velero/app/pkg/backup/item_backupper.go:326" error.function="github.com/vmware-tanzu/velero/pkg/backup.(*itemBackupper).executeActions" logSource="/remote-source/velero/app/pkg/backup/backup.go:417" name=<name>
```

#### Cause

An issue exists with the default OADP plug-in and it must be disabled to continue.

#### Resolution

Do the following steps to disable the plug-in:

1. In the OpenShift console, go to Administration > CustomerResourceDefinitions.
2. Search for the CustomResourceDefinition **DataProtectionApplication**.
3. In the Instances tab, locate the instance that is named **velero**.
4. Open the YAML file in edit mode for the instance.
5. Under the entry **spec:velero:defaultPlugins**, remove the line for **openshift**.
6. Save the YAML file.

## 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 Storage 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 Storage 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 Storage 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-bfy26 | 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-z1679              | 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.

## Known issues and limitations

- 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 virtual machine needs to be offline, or the hot pluggable volumes must be unmounted from the virtual machine for snapshots and backups to succeed. For more information, see [Hot-plugging VM disks](#).

## Restore issues

List of restore issues in Backup & Restore service of IBM Storage Fusion.

### 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 Storage 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 5 GB.

#### Resolution

Increase the IBM Storage Scale PVC size to a minimum of 5 GB and do a backup and restore operation.

## Cannot restore multiple namespaces to a single alternative namespace

---

You cannot restore multiple namespaces to a single alternative namespace. If you attempt such a restore, then the job fails. Example transaction manager log:

```
023-06-27 15:05:53,633 [TM_5] [c2a4b768-2acc-4169-9c5f-f88e60f3be2b] [restoreguardian:resre_application_w_recipe Line 172] [INFO] - alt!
```

```
2023-06-27 15:05:53,633 [TM_5] [c2a4b768-2acc-4169-9c5f-f88e60f3be2b] [restoreguardian:restore_application_w_recipe Line 176] [ERROR] -
```

## 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 Storage 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 Storage 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, it can eventually delete the configuration file and subdirectories that are not yet modified. All restic commands check this configuration file to determine the initialization of the restic repository. If it does not exist, the restore jobs fail to find the repository and restore the backup. Subsequent backup jobs cannot find the repository, and initialization attempts can result in a repository with errors.

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 VirtualMachineSnapshot 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,volumesnapshots.snapshot.storage.k8s.io,datauploads.velero.io,persistentvolumes,persistentvolumeclaims,serviceaccounts,secrets,configmaps,limitranges,pods,replicasesets.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"
```

### Problem statement

Datamover operator must restore CephFS and IBM Storage Scale 5.2.0+ snapshots for backup by using the **ReadOnlyMany** access modes.

### Resolution

## Backup & Restore service install and upgrade issues

---

Use this troubleshooting information to resolve install and upgrade problems that are related to Backup & Restore service.

## Application information not repopulated after service re-installation

---

### Problem statement

The application information gets removed during Backup & Restore service uninstallation. It does not get repopulated in MongoDB after the service re-installation. 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 Storage 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 Storage 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 navigate 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 Storage 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 Crashloopbackoff state.

#### Resolution

In the resource settings of `guardian-dp-operator` pod that resides in `ibm-backup-restore` namespace, set the value of IBM Storage Fusion operator memory limits to 1000Mi.

Example:

```
resources:
 limits:
 cpu: 1000m
 memory: 1000Mi
```

```
requests:
 cpu: 500m
 memory: 250Mi
```

## storage-operator pod crashes with CrashLoopBackOff status

---

### Problem statement

The storage-operator pod crashes due to OOM error.

### Resolution

To resolve the error, increase the memory limit from 300Mi to 500Mi.

1. Log in to the Red Hat® OpenShift® web console as an administrator.
2. Select the project `ibm-spectrum-fusion-ns`.
3. Select the `isf-storage-operator-controller-manager-xxxx` pod, and go to the YAML tab.
4. In the YAML, change the memory limit for `isf-storage-operator-controller-manager-xxxx` container from 300Mi to 500Mi.

```
containers:
 - resources:
 limits:
 cpu: 100m
 memory: 500Mi
```

5. Click Save.

## Backup & Restore service goes into unknown state

---

### Problem statement

This Backup & Restore service might goes into unknown state when you upgrade IBM Storage Fusion from 2.6.x to 2.8.0.

### Resolution

It automatically shows healthy on the IBM Storage Fusion user interface after you upgrade the Backup & Restore service.

## MongoDB pod crashes with CrashLoopBackOff status

---

### Problem statement

The MongoDB pod crashes due to 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 stuck at 5% during upgrade

---

### Diagnosis

1. In OpenShift user interface, go to the Installed Operator and filter on `ibm-backup-restore` namespace.
2. Click **IBM Storage Fusion Backup and Restore Server**.
3. Go to Subscriptions.
4. If you see the following errors, then do the workaround steps to resolve the issue:

```
"error validating existing CRs against new CRD's schema for "guardiancopyrestores.guardian.isf.ibm.com": error validating
```

### Resolution

1. From command line or command prompt, log in to the cluster and run the following commands to delete the `guardiancopyrestore` CRs and 2.6.0 cs:  

```
oc -n ibm-backup-restore delete guardiancopyrestore.guardian.isf.ibm.com --all
oc -n ibm-backup-restore delete csv guardian-dm-operator.v2.6.0
```
2. From OpenShift Container Platform console, go to Installed Operator and filter on `ibm-backup-restore` namespace.
3. Click **IBM Storage Fusion Backup and Restore Server**.
4. Go to Subscriptions.
5. Find the failing `installplan` and delete it.
6. Go to Installed Operator and go to `ibm-backup-restore` namespace.
7. Find **IBM Storage Fusion Backup and Restore Server** and click Upgrade available and approve the Install plan for **IBM Storage Fusion Backup and Restore Server**.
8. Wait for the service upgrade to resume.

## Backup & Restore service installation gets stuck after upgrade to IBM Storage Fusion 2.7.0

---

### Problem statement

If the installation or upgrade of the Backup & Restore service does not reach 100% completion, then check for failed startup probes. Run the following command to look for any pods that are not in the READY state:

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

Example output:

| NAME                            | READY | STATUS  | RESTARTS      | AGE |
|---------------------------------|-------|---------|---------------|-----|
| applicationsvc-855746ffbf-2pmz7 | 0/1   | Running | 5 (2m8s ago)  | 17m |
| ...                             |       |         |               |     |
| job-manager-845dc56b8d-r5w6j    | 0/1   | Running | 5 (2m50s ago) | 17m |
| ...                             |       |         |               |     |

1. Go to OpenShift Container Platform.
2. Go to the Installed Operators view with the selected `ibm-backup-restore` project or namespace.
3. Click IBM Storage Fusion Backup & Restore Server.
4. Select the Data Protection Server tab and click the instance.
5. Select the YAML tab and update the following:

```
triggerUpgrade: false
to
triggerUpgrade: true
```

6. Save and reload the YAML.

After some time, all of the pods must be READY and the Backup & Restore service must show healthy in IBM Storage Fusion user interface.

#### Diagnosis

If found, then describe each of the pods and look at the event section to determine if there is a startup probe failure.

Example output showing probe failure:

| Events: |           |                       |         |                                                          |
|---------|-----------|-----------------------|---------|----------------------------------------------------------|
| Type    | Reason    | Age                   | From    | Message                                                  |
| ---     | ---       | ---                   | ---     | -----                                                    |
| ...     |           |                       |         |                                                          |
| Warning | Unhealthy | 3m40s (x5 over 6m40s) | kubelet | Startup probe failed: HTTP probe failed with statuscode: |

If one or more pods show startup probe failures, then patch the probes to provide additional start time. Update the following script and run to patch the probes. To determine the respective deployment names, update the DEPLOYMENT\_NAMES variable with the list of deployments that have failing startup probes and run the script.

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

Example output:

| NAME           | READY | UP-TO-DATE | AVAILABLE | AGE |
|----------------|-------|------------|-----------|-----|
| ...            |       |            |           |     |
| applicationsvc | 0/1   | 1          | 1         | 12h |
| ...            |       |            |           |     |
| job-manager    | 0/1   | 1          | 1         | 12h |
| ...            |       |            |           |     |

Now, check whether the pods are online. If the startup probes continue to fail, then increase the STARTUP\_DELAY\_SEC parameter and try again.

If it is an upgrade and the pods come online, but the progress does not reach 100% in the IBM Storage Fusion user interface even after 30 minutes, then you must update the Backup & Restore Server CR to re-trigger the upgrade.

#### Resolution

Update the Backup & Restore Server CR to re-trigger the upgrade

## Backup & Restore service install 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 Storage 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-minio
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.

## IBM Cloud Pak for Data backup and restore issues

This section lists the troubleshooting tips and tricks during backup and restore of IBM Cloud Pak for Data.

## Original namespace as destination for IBM Cloud Pak for Data application

#### Problem statement

OpenShift® restore from IBM Storage Protect Plus does not force you to select the original namespace as destination for IBM Cloud Pak for Data application.

#### Impact

From IBM Storage Protect Plus user interface, IBM Cloud Pak for Data restore fails whenever you do not select the original namespace as the target destination to restore to.

#### Resolution

Whenever you restore IBM Cloud Pak for Data 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 Cloud Pak for Data applications. For more information about the procedure, see [https://www.ibm.com/docs/en/SSQNUZ\\_4.5/test/cpd/admin/online\\_bar\\_fusion\\_spp.html](https://www.ibm.com/docs/en/SSQNUZ_4.5/test/cpd/admin/online_bar_fusion_spp.html).

## Tethered namespace issues

---

#### Problem statement

For IBM Cloud Pak for Data Application type, you can tether secondary or other namespaces to the primary IBM Cloud Pak for Data platform namespace. However, after a successful restore, the IBM Storage Fusion application associated with the IBM Cloud Pak for Data 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 Cloud Pak for Data 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.

## 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: BootstrapSecretAvailable
 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` and 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

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 Storage 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 2.8 on IBM Storage Fusion 4.16, 4.15 and 4.14 releases. 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 Storage Fusion operator is installed.

## Troubleshooting Global Data Platform issues

Warning: Do NOT delete IBM Storage Scale pods. Deletion of Scale pods in many circumstances has implications on availability and data integrity.

## 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: | <b>False</b>   |
| Type:   | <b>Success</b> |

#### 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 Storage 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.

## Troubleshooting IBM Fusion Data Foundation service

Use these troubleshooting information to know the problem and workaround when install or configure IBM Fusion Data Foundation service.

- [Known issues and limitations](#)

Known issues and limitations in IBM Fusion Data Foundation.

## IBM Fusion Data Foundation service error scenarios

Use these troubleshooting information to know the problem and workaround when install or configure IBM Fusion Data Foundation service.

- [Red Hat OpenShift Data Foundation storage node failure](#)
- [Red Hat OpenShift Data Foundation Object Storage Device \(OSD\) failure](#)
- [Local storage operator unable to find candidate storage nodes](#)
- [IBM Fusion Data Foundation capacity cannot be loaded](#)
- [IBM Fusion Data Foundation cluster fails due to pending StorageClusterPreparing stage](#)

## Local storage operator unable to find candidate storage nodes

#### Problem statement

When you configure a IBM Fusion Data Foundation cluster, you do not find any candidate storage nodes.

#### Cause

When you configure IBM Fusion Data Foundation cluster, only compute nodes with available disks (SSD/NVMe or HDD) get displayed in the Data Foundation page of IBM Storage 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

- In Red Hat OpenShift Container Platform console, go to Operators > Installed Operators.
- Verify whether the `LocalStorage` operator is installed successfully.
- Run the following command to get all the worker nodes:

```
oc get node -l node-role.kubernetes.io/worker=
```

- Run the following command to check if discovery results are created for all worker nodes.

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

- Run the following command to confirm that none of the nodes have a IBM Fusion Data Foundation storage label:

```
oc get node -l cluster.ocs.openshift.io/openshift-storage=
```

Note: In Linux on IBM zSystems 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 above checks pass, but the node still could not be seen in the IBM Storage Fusion user interface, then contact [IBM support](#).

## IBM Fusion Data Foundation capacity cannot be loaded

If you encounter this issue in the Data foundation page of IBM Storage Fusion user interface, then contact [IBM support](#).

## IBM 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-storagecluster" 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
```

## Red Hat OpenShift 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 Red Hat® OpenShift® Data Foundation storage node.

### Before you begin

Red Hat recommends that replacement nodes are 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 Red Hat OpenShift Data Foundation storage node failure and identify the failed node:

1. Set the Red Hat OpenShift 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 Storage 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-r5bxz 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-r5bxz
 - 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-r5bxz 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 resource
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, openshift
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.

- Create a new compute node and ensure that the new node is in ready state.
- 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-r5bxz
 - 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-r5bxz | 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 |
|-------------------|-----|-----|--------|-----------|
|-------------------|-----|-----|--------|-----------|

- 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-ls651 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-1k9g2 2/2 Running 0 5m18s
rook-ceph-mon-d-6945bbfc5-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-r5bx 2/2 <none>
rook-ceph-osd-2-5c5d48ff7c-sdd71 2/2 Running 0 5m17s
10.129.4.125 f09-prc4m-worker-cluster-c-mfb77 <none>
rook-ceph-osd-prepare-1bf7dd3d71fe899383e625dd0c27ea37-x9vtk 0/1 Completed 0 4h8m
10.128.6.79 f09-prc4m-worker-cluster-d-r5bx 2/2 <none>
rook-ceph-osd-prepare-24272b5641dc95baffc7932d78894e3c-zhz8m 0/1 Completed 0 5m24s
10.129.6.48 f09-prc4m-worker-cluster-b-djplp <none>
rook-ceph-osd-prepare-6f3c3b4626fec9888b6fbe5597af5d55ea-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-0-clwx-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 Storage Fusion user interface and check the health of the Storage cluster in the Health section.

## Red Hat OpenShift Data Foundation Object Storage Device (OSD) failure

For any kind of failed storage devices on the clusters backed by local storage devices, you must replace the Red Hat® OpenShift® Data Foundation Object Storage Device (OSD).

If you encounter this issue, contact [IBM support](#).

## Before you begin

Red Hat recommends that replacement OSD devices are configured with similar infrastructure and resources to the device being replaced. You can replace an OSD in Red Hat OpenShift Data Foundation deployed using local storage devices on the following infrastructures:

- Bare Metal
- VMware with local deployment
- SystemZ
- IBM Power Systems

## Procedure

Do the following steps to check for the occurrence of Red Hat OpenShift Data Foundation OSD failure:

1. Set the Red Hat OpenShift Data Foundation cluster to maintenance:

```
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 OSD:

check whether the OSD failed by using any of the following methods:

- Log in to Red Hat OpenShift Container Platform web console and go to your storage system details page.
- In the Overview > Block and File tab, check the Status section for any warning in the Storage cluster.
- If the warnings indicate OSD down or degraded, then contact [IBM support](#) to replace the Red Hat OpenShift Data Foundation failed OSD for your storage node in an internal-attached environment.

Example warning message:

```
1 osds down
1 host (1 osds) down
Degraded data redundancy: 333/999 objects degraded (33.333%), 81 pgs degraded
```

- Log in to IBM Storage Fusion user interface.
- Go to Data foundation page and check for warnings in the Health section for storage cluster.

Alternatively, you can use the `oc` command to identify the OSD:

```
oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

Sample output:

```
[root@fu40 ~]# oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED
NODE READINESS GATES
rook-ceph-osd-0-6c99fc999b-2s9mr 1/2 CrashLoopBackOff 5 (17s ago) 17m 10.128.4.216 fu49 <none>
<none>
rook-ceph-osd-1-764f9cff48-6gkg9 2/2 Running 0 16m 10.131.2.18 fu47 <none>
<none>
rook-ceph-osd-2-5d9d5984dc-8gkrz 2/2 Running 0 16m 10.129.2.53 fu48 <none>
<none>
```

In this example, `rook-ceph-osd-0-6c99fc999b-2s9mr` needs to be replaced and `fu49` is the Red Hat OpenShift Container Platform node on which the OSD is scheduled. And the failed OSD id is 0.

You can view the OSD details as well `ceph osd df` in the Ceph tools. And the failed OSD id is the same as in previous step.

3. Scale down the OSD deployment

Scale down the OSD deployment replica to 0

Verify the OSD id from previous step, the `rook-ceph-osd-0-6c99fc999b-2s9mr` and pod id is 0.

```
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:

```
[root@fu40 ~]# osd_id_to_remove=0
[root@fu40 ~]# oc scale -n openshift-storage deployment rook-ceph-osd-${osd_id_to_remove} --replicas=0
deployment.apps/rook-ceph-osd-0 scaled
```

Waiting to the `rook-ceph-osd` pod is terminated

Run the `oc` command to terminate `rook-ceph-osd` pod.

```
oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
```

Example output:

```
[root@fu40 ~]# oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
NAME READY STATUS RESTARTS AGE
rook-ceph-osd-0-6c99fc999b-2s9mr 0/2 Terminating 6 20m
```

Note: If the `rook-ceph-osd` pod is in terminating state and taking more time, then use the force option to delete the pod.

```
oc delete -n openshift-storage pod rook-ceph-osd-0-6c99fc999b-2s9mr --grace-period=0 --force
```

Example output:

```
[root@fu40 ~]# oc delete -n openshift-storage pod rook-ceph-osd-0-6c99fc999b-2s9mr --grace-period=0 --force
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The
resource may continue to run on the cluster indefinitely.
pod "rook-ceph-osd-0-6c99fc999b-2s9mr" force deleted
```

Verify whether `rook-ceph-osd` is terminated.

```
[root@fu40 ~]# oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
No resources found in openshift-storage namespace.
```

4. Remove the old OSD from the cluster.

Delete any old `ocs-osd-removal` jobs

Run the `oc` command to delete `ocs-osd-removal` jobs.

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

Remove the old OSD from the cluster

Ensure that you set the correct `osd_id_to_remove`.

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.

- More than three OSDs

```
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 -
```

- Only three OSDs or insufficient space (force delete)

```
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 -
```

Example output:

```
[root@fu40 ~]# echo $osd_id_to_remove
0
```

Verify the OSD is removed

Wait for the `ocs-osd-removal-job` pod is completed.

```
[root@fu40 ~]# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
NAME READY STATUS RESTARTS AGE
ocs-osd-removal-job-s4vhc 0/1 Completed 0 24s
```

Double confirm the logs.

```
[root@fu40 ~]# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
2022-11-25 16:08:49.858109 I | cephosd: completed removal of OSD 0
```

The PVC will go to `Pending`, and the `pv` will be `Released`.

```
openshift-storage ocs-deviceset-ibm-spectrum-fusion-local-0-data-3nsk8j Pending
ibm-spectrum-fusion-local 7m16s

local-pv-a2879220 600Gi RWO Delete Released openshift-
storage/ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b ibm-spectrum-fusion-local 41m
```

To locate the worker node, use the `oc` command to describe `pv`.

For example, `pv` host name is `fu49 kubernetes.io/hostname=fu49`.

```
[root@fu40 ~]# oc describe pv local-pv-a2879220
Name: local-pv-a2879220
Labels: kubernetes.io/hostname=fu49
Annotations: pv.kubernetes.io/bound-by-controller: yes
 pv.kubernetes.io/provisioned-by: local-volume-provisioner-fu49-96f64c0f-e5ed-4bb1-b4ff-
cad610562f58
 storage.openshift.com/device-id: scsi-36000c2913ba6a22c66120c73cb1edae6
 storage.openshift.com/device-name: sdb
Finalizers: [kubernetes.io/pv-protection]
StorageClass: ibm-spectrum-fusion-local
Status: Released
Claim: openshift-storage/ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b
Reclaim Policy: Delete
Access Modes: RWO
VolumeMode: Block
Capacity: 600Gi
Node Affinity:
 Required Terms:
 Term 0: kubernetes.io/hostname in [fu49]
Message:
Source:
Type: LocalVolume (a persistent volume backed by local storage on a node)
Path: /mnt/local-storage/ibm-spectrum-fusion-local/scsi-36000c2913ba6a22c66120c73cb1edae6
Events:
 Type Reason Age From Message
 ---- ---- -- ---- ---
 Warning VolumeFailedDelete 6m2s (x26 over 12m) deleter Error cleaning PV "local-pv-a2879220": failed to
get volume mode of path "/mnt/local-storage/ibm-spectrum-fusion-local/scsi-36000c2913ba6a22c66120c73cb1edae6":
Directory check for "/mnt/local-storage/ibm-spectrum-fusion-local/scsi-36000c2913ba6a22c66120c73cb1edae6"
failed: open /mnt/local-storage/ibm-spectrum-fusion-local/scsi-36000c2913ba6a22c66120c73cb1edae6: no such file
or directory
```

Note: If the `ocs-osd-removal-job` fails and the pod is not in the expected completed state, check the pod logs for further debugging.

Remove Encryption related configuration

Remove the **dm-crypt** managed device-mapper mapping from the OSD devices that are removed from the respective Red Hat OpenShift Data Foundation nodes if encryption was enabled during installation.

- For each of the previously identified nodes, do the following:

```
oc debug node/<node name>
chroot /host
dmsetup ls| grep <pvc name>
```

- Remove the mapped device.

```
cryptsetup luksClose --debug --verbose ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt
```

Example output:

```
[root@fu40 ~]# oc debug nodes/fu49
Starting pod/fu49-debug ...
To use host binaries, run `chroot /host`
If you don't see a command prompt, try pressing enter.
sh-4.4# chroot /host
sh-4.4# dmsetup ls
ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt (253:0)
sh-4.4# cryptsetup luksClose --debug --verbose ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt

cryptsetup 2.3.3 processing "cryptsetup luksClose --debug --verbose ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt"
Running command close.
Locking memory.
Installing SIGINT/SIGTERM handler.
Unblocking interruption on signal.
Allocating crypt device context by device ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt.
Initialising device-mapper backend library.
dm version [opencount flush] [16384] (*1)
dm versions [opencount flush] [16384] (*1)
Detected dm-ioctl version 4.43.0.
Detected dm-crypt version 1.21.0.
Device-mapper backend running with UDEV support enabled.
dm status ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt [opencountnoflush]
[16384] (*1)
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-1m227b-block-dmcrypt disappeared.
dm versions [opencount flush] [16384] (*1)
dm table ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt [opencount flush
securedata] [16384] (*1)
dm versions [opencount flush] [16384] (*1)
dm deps ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt [opencount flush]
[16384] (*1)
LUKS device header not available.
Deactivating volume ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt.
dm versions [opencount flush] [16384] (*1)
dm status ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmrypt [opencountnoflush]
[16384] (*1)
dm versions [opencount flush] [16384] (*1)
dm table ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt [opencount flush
securedata] [16384] (*1)
dm versions [opencount flush] [16384] (*1)
dm deps ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt [opencount flush]
[16384] (*1)
dm versions [opencount flush] [16384] (*1)
dm table ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmcrypt [opencount flush
securedata] [16384] (*1)
dm versions [opencount flush] [16384] (*1)
Udev cookie 0xd4d9390 (semid 0) created
Udev cookie 0xd4d9390 (semid 0) incremented to 1
Udev cookie 0xd4d9390 (semid 0) incremented to 2
Udev cookie 0xd4d9390 (semid 0) assigned to REMOVE task(2) with flags DISABLE_LIBRARY_FALLBACK
(0x20)
dm remove ocs-deviceset-ibm-spectrum-fusion-local-0-data-1m227b-block-dmrypt [opencount flush
retryremove] [16384] (*1)
Udev cookie 0xd4d9390 (semid 0) decremented to 1
Udev cookie 0xd4d9390 (semid 0) waiting for zero
```

Find the persistent volume (PV) that need to be deleted

Run the **oc** command to find the failed **PV**.

```
oc get pv -l kubernetes.io/hostname=<failed-osds-worker-node-name>
```

Example output:

```
[root@fu40 ~]# oc get pv -l kubernetes.io/hostname=fu49
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM
STORAGECLASS REASON AGE
local-pv-a2879220 600Gi RWO Delete Released openshift-storage/ocs-deviceset-ibm-
spectrum-fusion-local-0-data-1m227b ibm-spectrum-fusion-local 55m
```

Delete the released persistent volume (PV)

Run the **oc** command to delete the released **PV**.

```
oc delete pv <pv_name>
```

Example output:

```
[root@fu40 ~]# oc delete pv local-pv-a2879220
persistentvolume "local-pv-a2879220" deleted
```

5. Add new OSD into the node.

Add a new device physically to the node.

Track the provisioning of persistent volume (PV)s for the devices that match the deviceInclusionSpec.

It can take a few minutes to provision the PVs. Once the PV is identified, it adds itself to the cluster automatically.

- lvs spec

```
oc -n openshift-local-storage describe localvolumeset ibm-spectrum-fusion-local
```

Example output:

```
...
Spec:
Device Inclusion Spec:
 Device Types:
 disk
 part
 Max Size: 601Gi
 Min Size: 599Gi
Node Selector:
 Node Selector Terms:
 Match Expressions:
 Key: cluster.ocs.openshift.io/openshift-storage
 Operator: In
 Values:
```

Delete the **ocs-osd-removal-job**

Run the **oc** command to delete the **ocs-osd-removal-job**.

```
```
oc delete -n openshift-storage job ocs-osd-removal-job
```
```
[root@fu40 ~]# oc delete -n openshift-storage job ocs-osd-removal-job
job.batch "ocs-osd-removal-job" deleted
````
```

6. Verify that there is a new OSD running

Verify new OSD pod is running

Run the **oc** command to check the new OSD pod is running.

```
oc get -n openshift-storage pods -l app=rook-ceph-osd
```

Example output:

```
[root@fu40 ~]# oc get -n openshift-storage pods -l app=rook-ceph-osd
NAME READY STATUS RESTARTS AGE
rook-ceph-osd-0-7f99b8cccd5-ssj5w 2/2 Running 0 7m31s <-- This pod
rook-ceph-osd-1-764f9cff48-6gkg9 2/2 Running 0 64m
rook-ceph-osd-2-5d9d5984dc-8gkrz 2/2 Running 0 64m
```

Tip: If the new OSD does not show as Running after a few minutes, restart the **rook-ceph-operator** pod to force a reconciliation.

```
oc delete pod -n openshift-storage -l app=rook-ceph-operator
```

Verify new PVC is created

Run the **oc** command to check whether the pods are running.

```
oc get pvc -n openshift-storage
```

Example output:

```
[root@fu40 ~]# oc get pvc -n openshift-storage
NAME STATUS VOLUME
CAPACITY ACCESS MODES STORAGECLASS
db-noobaa-db-pg-0 Bound pvc-783036b5-ec40-41a7-91e5-9e179fd24cc3 50Gi
RWO ocs-storagecluster-ceph-rbd 65m <--This one
ocs-deviceset-ibm-spectrum-fusion-local-0-data-04vwvq Bound local-pv-b45b1d67
600Gi RWO ibm-spectrum-fusion-local 66m
ocs-deviceset-ibm-spectrum-fusion-local-0-data-24nj5t Bound local-pv-c3de9110
600Gi RWO ibm-spectrum-fusion-local 66m
ocs-deviceset-ibm-spectrum-fusion-local-0-data-3nsk8j Bound local-pv-1c9f3b11
600Gi RWO ibm-spectrum-fusion-local 34m
[root@fu40 ~]#
[root@fu40 ~]# oc get pv
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM
STORAGECLASS REASON AGE
local-pv-1c9f3b11 600Gi RWO Delete Bound openshift-
storage/ocs-deviceset-ibm-spectrum-fusion-local-0-data-3nsk8j ibm-spectrum-fusion-local 10m
<--This one
local-pv-b45b1d67 600Gi RWO Delete Bound openshift-
storage/ocs-deviceset-ibm-spectrum-fusion-local-0-data-04vwvq ibm-spectrum-fusion-local 68m
local-pv-c3de9110 600Gi RWO Delete Bound openshift-
storage/ocs-deviceset-ibm-spectrum-fusion-local-0-data-24nj5t ibm-spectrum-fusion-local 68m
pvc-783036b5-ec40-41a7-91e5-9e179fd24cc3 50Gi RWO Delete Bound openshift-
storage/db-noobaa-db-pg-0 ibm-spectrum-fusion-local 65m
```

Verify the OSD Encryption settings

If cluster wide encryption is enabled, ensure that the **crypt** keyword is next to the **ocs-deviceset** name.

```
oc debug node/<new-node-name> -- chroot /host lsblk -f
oc debug node/<new-node-name> -- chroot /host dmsetup ls
```

Example output:

```
[root@fu40 ~]# oc debug node/fu49 -- chroot /host lsblk -f
Starting pod/fu49-debug ...
To use host binaries, run `chroot /host`
NAME MOUNTPOINT FSTYPE LABEL
UUID crypto_LUKS pvc_name=ocs-deviceset-ibm-
loop1 spectrum-fusion-loca 6a8244eb-55d6-48cc-8e68-33436e512bc6
loop2 spectrum-fusion-loca fa228ec1-0b1d-43ad-8707-9ecd38bfb1f8
sda
|-sda1
|-sda2 vfat EFI-SYSTEM
A084-4057
|-sda3 ext4 boot
`-sda4
7d757098-d548-4b7b-8c9a-3dd4f34ceca1 /boot xfs root
1cd39805-6936-458d-ae8c-39313bb71c95 /sysroot
sdc spectrum-fusion-loca fa228ec1-0b1d-43ad-8707-9ecd38bfb1f8
`-ocs-deviceset-ibm-spectrum-fusion-local-0-data-3nsk8j-block-dmcrypt
sr0

Removing debug pod ...
[root@fu40 ~]# oc debug node/fu49 -- chroot /host dmsetup ls
Starting pod/fu49-debug ...
To use host binaries, run `chroot /host`
ocs-deviceset-ibm-spectrum-fusion-local-0-data-3nsk8j-block-dmrypt (253:0)

Removing debug pod ...
```

Note: If verification steps fail, then contact Red Hat support.

Exit maintenance mode

Run the **oc** command to 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
```

7. Go to Data foundation page in IBM Storage Fusion user interface and check the health of the Storage cluster in the Health section.

---

## Known issues and limitations

Known issues and limitations in IBM Fusion Data Foundation.

The IBM Fusion Data Foundation dashboard includes inaccurate capacity information on the amount of data that is written to CephFS PVs. For more information about the issue, see [https://bugzilla.redhat.com/show\\_bug.cgi?id=2252911](https://bugzilla.redhat.com/show_bug.cgi?id=2252911).

---

## IBM Storage Fusion Data Cataloging known issues

List of all troubleshooting and known issues exist in version 2.1 of Data Cataloging.

The following known issues exist in 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](#).

Note: The issues [Data Cataloging login page stuck in loading](#) and [Data Cataloging service does not show full menu because user have not assigned the correct roles](#) are resolved and not applicable in 2.7.1 IBM Storage Fusion release.

---

## Data Cataloging service in Metro-DR setup shows in Degraded state

Diagnosis

1. 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 *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-zookeeper 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 Data Cataloging. After successful upgrade of 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 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}')`
```

Simply 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')`
```

## Querying available applications on Docker Hub is not working

When you retrieve the list of available applications on Docker Hub by using the public registry endpoint, the query retrieves an empty response:

```
$ tcurl https://${OVA}/api/application/appcatalog/publicregistry | jq
% Total % Received % Xferd Average Speed Time Time Current
 Dload Upload Total Spent Left Speed
100 78 100 78 0 0 135 0 --:--:-- --:--:-- 135
{
 "success": "false",
 "message": "Could not retrieve available applications."
}
```

To avoid this issue, you need to open a browser and access the following URL:[Docker Documentation](#).

The above link retrieves the list of Data Cataloging applications available in the public registry. The image name of the application that is selected from the query output can be used to create a JSON file with the information that is needed to run the application, as shown in the following:[Spectrum Discover Documentation](#).

## Running applications from the catalog

Querying and running available applications from the catalog. Currently, the REST API public registry endpoint to retrieve the list of available applications from the DockerHub is not working. For that reason, the Data Cataloging application catalog is only available in the following repository:[Spectrum Discover App Catalog](#)

## Scale Live Events do not get populated due to the timestamp field value being invalid

### Problem statement

Live events for IBM Storage Scale connections do not work as expected. After the initial scan of an IBM Storage Scale connection, only file deletions will result in a live update of the files that were discovered by the scan. If a file is modified or added, the live update will fail, and there will be no change reflected in the Data Cataloging product. This error is recorded by the DB2 product and can be surfaced by extracting the bad updates and the corresponding log files from one of the db2 pods. The following script executed within one of the db2 pods would extract these errors for analysis.

```
cd /mnt/bluemat0/home/bluadmin
sudo ls *.bad > /tmp/output.bad
sed 's/bad/log/' /tmp/output.bad > /tmp/output.log
sudo zip /tmp/output.zip -r . -i@/tmp/output.bad -i@/tmp/output.log
rm /tmp/output.bad
rm /tmp/output.log
cd /tmp
```

### Resolution

The workaround is to perform schedule scans of the IBM Storage Scale connections so that all file changes are up to date.

## Data Cataloging login page stuck in loading

This issue can be triggered by the following two possible causes:

Discover GUI is inaccessible after nodes put in maintenance mode and moved back to ready state for short period in OpenShift Container Platform, but service status remains healthy.

Follow the steps to resolve this issue:

1. Edit the `isd-ui-backend` deployment to change `OCPAPI_SERVER` `envvar` current value to use the new value.  
`kubernetes.default.svc.cluster.local`
2. Wait for OpenShift to create a pod with the new deployment and delete the current one.
3. Review whether the changes are applied in the Environment tab.
4. Open the Data Cataloging service, clean the cookies, and reload the user interface.

The Data Cataloging login page is stuck in loading when trying to access the user interface after you finish the installation using SSO login in Baremetal cluster.

As a workaround, disable the SSO authentication. For procedure, see [Disabling SSO login](#).

## Adding S3 connection gives false negative

### Problem statement

When a connection of type S3 is added through 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.

## When installation is at 80%, up to six pods might experience a Crash-loop

### Back-off error

### Problem statement

This issue happens when pods are waiting for the db-schema pod to finish the internal schema upgrades.

### Resolution

When the db-schema pod goes into "Running" state, after about 6 restarts, the pods go into running state, and installation completes successfully.

## Scale datamover AFM and ILM capabilities not working properly due to SDK misleading function when deploying an application

---

### Problem statement

When deploying Data Cataloging service deployments pods `scaleafmdata mover` and `scaleilmdata mover` 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-data mover (exit status 1; not expected)
2023-07-20 02:51:55,368 INFO gave up: scaleafm-data mover 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

Once 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
```

## Data Cataloging service does not show full menu because user have not assigned the correct roles

---

### Cause

Data Cataloging user interface does not show the side bar menu correctly because user have not assigned the correct roles.

### Resolution

Run the following steps to resolve this issue:

1. Run the following command to delete the current user in the Data Cataloging service.

```
oc -n <dcs_name_space> exec -c spectrum-discover deployment/isd-keystone -- /bin/sh -c "source /keystone_sdadminrc && openstack user delete <user_name>"
```

Important: Repeat the step 1 again if you get the following error.

```
"/keystone_sdadminrc: No such file or directory"
you will need to create first the file with the following:
```

```
echo -e 'export OS_IDENTITY_API_VERSION="3"\nexport OS_AUTH_URL="http://127.0.0.1:5000"\nexport
OS_USER_DOMAIN_ID="default"\nexport OS_PROJECT_DOMAIN_ID="default"\nexport OS_PROJECT_NAME="spectrum-
discover"\nexport OS_USERNAME="sdadmin"\nexport OS_PASSWORD="Passw0rd"' > /keystone_sdadminrc
```

2. Run the following command to add CRB cluster-admin to the user in OpenShift Container Platform.

```
oc adm policy add-cluster-role-to-user cluster-admin <user_name>
```

3. Clear cache and re-launch OpenShift Container Platform service.

## Data Cataloging service goes degraded state after IBM Storage Fusion HCI System rack restart

---

### Problem statement

The Data Cataloging service goes degraded after some of the nodes are restarted or after IBM Storage Fusion HCI System 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 Data Cataloging service must be in a healthy state after all the nodes are backed up.

## Data Cataloging service is unhealthy due to etcd crash

---

#### Problem statement

The Data Cataloging service goes into an unhealthy state due to `etcd` crash.

#### Diagnosis

Run the following steps to diagnose this issue:

1. Run the following command to check `etcd` pods that are not running.

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

2. Run the following command to check whether the logs show this error `error": "wal: crc mismatch`.

```
oc -n ibm-data-cataloging logs -l component=etcd
```

#### Resolution

Run the following steps to resolve this issue:

1. Run the following command to scale down `etcd`.

```
oc -n ibm-data-cataloging scale --replicas=0 sts/c-isd-etcd
```

2. Run the following command to remove the affected `etcd` files.

```
oc -n ibm-data-cataloging rsh c-isd-db2u-0
```

3. Run the following commands to refresh the `etcd`.

```
sudo rm -rf /mnt/blumeta0/etcd/c-isd-etcd-0/default.etcd
sudo mv /mnt/blumeta0/etcd/c-isd-etcd-0/member_id /mnt/blumeta0/etcd/c-isd-etcd-0/member_id.bk
exit
```

4. Run the following command to scale up `etcd`.

```
oc -n ibm-data-cataloging scale --replicas=1 sts/c-isd-etcd
```

5. Run the following command to restart Db2.

```
oc -n ibm-data-cataloging rsh c-isd-db2u-0
```

6. Run the following command to restart Db2 high availability system.

```
sv stop wolverine
sv start wolverine
exit
```

## Db2 license does not display correctly on the upgrade set up

---

#### Problem statement

The Db2 license displays incorrectly on the Data Cataloging service upgrade set up.

#### Resolution

Run the following steps to resolve this issue:

1. Run the following command to get the subsequent 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 -o yaml)
```

3. Once 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](#).

## Frequently asked questions

---

Commonly asked question and answers on deployment, storage, and backup of IBM Storage Fusion.

- **Install**  
This section answers questions related to installation of IBM Storage Fusion.
  - **Data Foundation Storage**  
This section answers questions that are related to storage in IBM Storage Fusion.
  - **Backup and Restore**  
This section answers questions related to backup and restore feature of IBM Storage Fusion.
  - **Data Cataloging**  
This section answers questions that are related to Data Cataloging in IBM Storage Fusion.
  - **Support and serviceability**  
This section answers questions related to serviceability.
- 

## Install

This section answers questions related to installation of IBM Storage Fusion.

1. What are the supported deployments?  
Yes, you can install IBM Storage Fusion on OpenShift® Container Platform that runs on On-premises VMware, On-premises Bare Metal, Linux on IBM zSystems, Microsoft Azure, Amazon Web Services, IBM Cloud®, and IBM Power Systems.
  2. What are IBM Storage Fusion services supported on the different deployment platforms for IBM Storage Fusion?  
For more information the support matrix, see [IBM Storage Fusion Services support matrix](#) matrix.
  3. What are the resource requirements for fusion software?  
To know about resource requirements, see [System requirements](#)
  4. Does IBM Storage Fusion support disconnected installation?  
Yes, it supports offline installation.
  5. Is Deployment Guide available for IBM Storage Fusion?  
The deployment guide is available in the following IBM Documentation location:[Deploying IBM Storage Fusion](#).
  6. IBM Storage Fusion can be deployed on UPI-installed OpenShift running on VMware? Does that imply that IBM Storage Fusion cannot be deployed if OpenShift is installed using IPI for VMware?  
IBM Storage Fusion assumes a bring-your-own OpenShift model. In other words, it assumes that OpenShift Container Platform is already installed. IBM Storage Fusion does not dictate how it would or should have been installed.
  7. Can IBM Storage Fusion run on AWS ROSA and IBM Cloud ROKS? Can IBM Storage Fusion run on public cloud (IBM Cloud, AWS) non-managed OpenShift environment?  
Yes, the IBM Storage Fusion runs on AWS ROSA and IBM Cloud ROKS and also on public cloud (IBM Cloud, AWS) non-managed OpenShift environment.
- 

## Data Foundation Storage

This section answers questions that are related to storage in IBM Storage Fusion.

1. Can I install the Data Foundation service to use HDD disks?  
Yes, the HDD is supported for development purpose. Use the SSD for production.
  2. Does the Data Foundation support Object Storage?  
Yes, the Data Foundation support Object Storage.
  3. What are the supported deployments?  
For supported deployments, see [IBM Storage Fusion Services support matrix](#).
  4. What is the supported Data Foundation version?  
The Data Foundation version is the same as the OpenShift® Container Platform version, that is, on OpenShift Container Platform 4.14, install Data Foundation 4.14.
  5. What is the difference between Fusion Data Foundation and Red Hat® OpenShift Data Foundation?  
The Fusion Data Foundation and Red Hat OpenShift Data Foundation have the same functions. The Fusion Data Foundation includes the Red Hat OpenShift Data Foundation Advanced subscription. The Fusion Data Foundation image is from the IBM image repository and the Red Hat OpenShift Data Foundation image from Red Hat image repository.
  6. What levels of encryption are supported?  
Both cluster-wide encryption and storageclass encryption are supported.
  7. How to handle and support key management?  
Support is available for external Key Management System, either HashiCorp Vault or Thales CipherTrust Manager.
  8. 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 Storage Fusion.

1. Can I install the Backup & restore as a day 2 operation?

Yes, from the Services user interface page of IBM Storage 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 Storage Fusion to create both local backups as well as copy data to an external, even offsite, S3 compliant object store.

4. Does IBM Storage Fusion backs up application data consistently across all the Persistent Volumes (PV)? Which applications are supported in an application consistent backup mode within IBM Storage 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 Storage Fusion, see [Achieving data consistency](#).

5. Will IBM Storage Fusion data protection support ODF remote mount?

Yes, ODF supports a remote mount. For more information, see [https://access.redhat.com/documentation/en-us/red\\_hat\\_openshift\\_data\\_foundation/4.15/html/deploying\\_openshift\\_data\\_foundation\\_in\\_external\\_mode/index](https://access.redhat.com/documentation/en-us/red_hat_openshift_data_foundation/4.15/html/deploying_openshift_data_foundation_in_external_mode/index)

---

## Data Cataloging

This section answers questions that are related to Data Cataloging in IBM Storage Fusion.

1. What is Data Cataloging?

Data Cataloging is an IBM Storage 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 and 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). Once terminated, you see your data that is indexed in the 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?

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 Data Cataloging?

Yes, IBM Storage Scale can be used as storage provider to install 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 Storage Fusion to connect your Fusion cluster with the remote Scale.

6. Can we use Red Hat® OpenShift® Data Foundation as storage provider for Data Cataloging?

Yes, Red Hat OpenShift Data Foundation can be used as storage provider for Data Cataloging. You just need to install the Data Cataloging service available in IBM Storage Fusion. Once installed, you can install Data Cataloging. When installing Data Cataloging, be sure that you select the `ocs-storagecluster-cephfs` as storage class.

7. Can we use any storage class to install Data Cataloging?

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.

---

## 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 Storage Fusion event currently in the event queue:

```
oc get events --field-selector reason=ISFEventManager
```

The following OC command lists the contents of each IBM Storage 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 Storage 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: <https://community.ibm.com/community/user/storage/blogs/byron-williams/2021/10/28/viewing-and-examining-logs-in-ibm-spectrum-fusion?CommunityKey=e596ba82-cd57-4fae-8042-163e59279ff3>

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 Storage Fusion Events page?

See the "Event UI" section of this blog post: <https://community.ibm.com/community/user/storage/blogs/byron-williams/2021/10/26/viewing-and-interpreting-events-in-ibm-spectrum-fu?CommunityKey=e596ba82-cd57-4fae-8042-163e59279ff3&tab=recentcommunityblogsdashboard>

12. What services are sold with IBM Storage 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 Storage 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 Storage 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 Storage Fusion was announced on 8/10/21. Support terms are available for 1, 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 Storage 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://worklodapid1.fyre.ibm.com:8081/spectrum\\_fusion/9155\\_spectrum\\_fusion\\_24\\_service\\_book.pdf](http://worklodapid1.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 Storage 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 Storage 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 Storage 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 Storage 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 Storage Fusion will support Premium Expert Care and will include the installation services.

27. Is Lab Services required to implement IBM Storage Fusion in the Customer location?

IBM Storage Fusion does not require lab services. IBM Storage 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 Storage 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 Storage 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 Storage Fusion online product documentation in IBM Documentation (IBM Docs) is enabled for accessibility. The accessibility features are described in [IBM Documentation](#) at [Accessibility](#).

IBM Storage Fusion uses the recent W3C Standard, [WAI-ARIA 1.0](#) ([www.w3.org/TR/wai-aria/](http://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](http://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/](http://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 Storage Fusion.

### Keyboard navigation

This product uses standard Microsoft Windows navigation keys and the following keyboard shortcuts.

Table 1. Keyboard shortcuts in IBM Storage 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 Storage Fusion user interfaces do not have content that flashes 2 - 55 times per second.

The IBM Storage 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 Storage 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 manager 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 Storage Fusion considerations for GDPR readiness

---

### Notice

---

This document is intended to help you in your preparations for GDPR readiness. It provides information about features of IBM Storage 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 Storage Fusion to help your organization with GDPR readiness.

## Data life cycle

---

### User accounts

The IBM Storage 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 Storage 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 Storage Fusion.

#### Personal data used for online contact with IBM

IBM Storage Fusion clients can submit online comments/feedback/requests to contact IBM about IBM Storage Fusion subjects in various ways, primarily:

- Public comments area on pages in the IBM Storage Fusion community.
- Public comments area on pages of IBM Storage Fusion documentation in IBM Documentation.
- Public comments in the IBM Storage Fusion space of dWAnswers
- Feedback forms in the IBM Storage 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 Storage 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 Storage 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 Storage 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 Storage Fusion.

## Data monitoring

---

IBM Storage 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 Storage Fusion meets the following data subject rights: right to access, modify, forgotten, and portability.

## Glossary

---

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

---

### 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.

### allocatable extent limit

A maximum total capacity for the system. The allocatable extent limit is calculated from pool extent sizes.

### 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

---

### 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 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](#).

### 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 <a href="#">CIM object manager</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| CKD                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                                    | See <a href="#">count key data</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| CLI                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                                    | See <a href="#">command-line interface</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| client                             | A software program or computer that requests services from a server. See also <a href="#">host</a> , <a href="#">server</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 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.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 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 <a href="#">data record</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| count key data (CKD)               | <ol style="list-style-type: none"> <li>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.</li> <li>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 <a href="#">data record</a>, <a href="#">storage architecture type</a>.</li> </ol> |
| CRU                                | See <a href="#">customer-replaceable unit</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 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.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 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 record              |                                                                                                                                                                                                                                                                                                                                                                 |
|                          | A basic unit of data recording format. See also <a href="#">count key data</a> , <a href="#">fixed-block architecture</a> .                                                                                                                                                                                                                                     |
| 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 <a href="#">data reduction savings</a> , <a href="#">stored capacity</a> .                                                                                         |
| 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 <a href="#">data reduction</a> .                                            |
| destage                  | To move data from cache to a nonvolatile storage medium.                                                                                                                                                                                                                                                                                                        |
| 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 <a href="#">rebuild area</a> . |
| DNS                      | See <a href="#">Domain Name System</a> .                                                                                                                                                                                                                                                                                                                        |
| 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.

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.

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](#).

## 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 in order 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.

#### HyperSwap

Pertaining to a function that provides continuous, transparent availability against storage errors and site failures, and is based on synchronous replication.

## I

---

#### 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](#).

## K

---

#### key server

1. A server that negotiates the values that determine the characteristics of a dynamic virtual private network (VPN) connection that is established between two endpoints.

## L

---

### 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

---

### 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.

## 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](#).

### 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.

## 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.

### reserved capacity

The amount of used capacity that is made up of capacity reserved for system use. See also [total usable physical capacity](#).

## S

---

### 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.

### 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 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.

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

---

#### 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

---

#### 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](#).

## 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

---

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\_.

## Programming interface information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of IBM Storage Fusion. This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM Storage 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](http://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 via 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](#).